Web程序员成功之路

CSS Web



- 迅速提高读者Web开发能力，全面挖掘读者开发潜力
- 一线资深Web程序员经验力作，窗内网独家推荐自学教材
- 11个小时视频教学，简化学习过程
- 112个实战案例与理论知识综合讲解，提高应用能力
- 网站互动教学（www.itzcn.com），QQ群在线帮助读者解疑

张芳芳　侯俊杰　编著





清华大学出版社

Web 程序员成功之路

CSS Web 开发学习实录

张芳芳　侯俊杰　编著

清华大学出版社
北　京

内 容 简 介

CSS 是用于控制网页样式，并允许将样式信息与网页内容分离的一种标记性语言。只需在 CSS 代码中做一些简单的修改，就可以改变同一页面中的不同部分，或者统一网站页面的外观和格式，例如颜色、边框、背景图片、透明度和大小等。

本书从介绍 CSS 技术的基础知识开始，围绕 Web 设计的各个方面予以展开，并通过大量实例深入剖析 CSS 应用的核心。本书分 20 章，主要内容包括 CSS 选择符、CSS 伪类、字体样式、文本样式、区块样式、列表样式、表格和表单样式以及特效滤镜。此外，还对 CSS 在页面布局、XML、JavaScript 以及 Dreamweaver 中的应用进行了介绍。最后使用 div+CSS 制作了两个综合实例。

本书适合具有一定 XHTML 基础，希望通过 CSS 提高 Web 设计技能的读者，同时也适合网页制作初学者学习，也可作为相关培训机构的参考用书。

图书在版编目(CIP)数据

CSS Web 开发学习实录/张芳芳，侯俊杰编著. --北京：清华大学出版社，2011.6

(Web 程序员成功之路)

ISBN 978-7-302-25577-2

Ⅰ. ①C…　Ⅱ. ①张…　②侯…　Ⅲ. ①网页制作工具，CSS　Ⅳ. ①TP393.092

中国版本图书馆 CIP 数据核字(2011)第 092729 号

责任编辑：张　瑜　杨作梅
装帧设计：杨玉兰
责任校对：李玉萍
责任印制：

出版发行：清华大学出版社　　**地　址：**北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　编：**100084
社 总 机：010-62770175　　**邮　购：**010-62786544
投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：190×260　**印　张：**35　**字　数：**853 千字
附光盘 1 张
版　　次：2011 年 6 月第 1 版　　**印　次：**2011 年 6 月第 1 次印刷
印　　数：1~4000
定　　价：68.00 元

产品编号：

前　　言

一个优秀的网站不仅需要具有一套完整的内容结构，还需要巧妙且美观的外观布局及合理的内容搭配。此外，网站的页面设计具有统一的风格同样是至关重要的，这样才能增强网站的吸引力，提高访问量，从而在众多的网站中脱颖而出。

CSS(Cascading Style Sheets，层叠样式表)就可以实现这种功能，使用 CSS 可以有效地对页面的布局、字体、颜色、背景和其他效果实施精确的控制。只需在 CSS 代码中做一些简单的修改，就可以改变同一页面中的不同部分，或者统一网站页面的外观和格式。而且可以使 HTML 文档代码更加简练，文档结构更加清晰，并能有效地缩短浏览器的加载时间。

本书内容

第 1 章　CSS 快速入门。本章首先介绍 CSS 的概念、历史和优点，接下来介绍样式表的使用方法、CSS 注释、CSS 的优先级和单位等信息，最后从诸多网页制作工具中挑选一款适合自己的网页制作工具。

第 2 章　选择符。本章详细讲解每种 CSS 选择符，像通用选择符、标签选择符、ID 选择符和类选择符等。

第 3 章　伪类和伪元素。本章将会详细讲解每个 CSS 伪类和伪元素的用法，像链接伪类、:focus、:first-child、:before 和:after 等。

第 4 章　字体属性。主要介绍如何对字体设置属性，像字体颜色、字体大小、字体外观、字体行高与间隔等。

第 5 章　文本属性。主要介绍 CSS 中的文本属性，像 text-indent、vertical-align、white-space 和 text-overflow 等。

第 6 章　边框与尺寸属性。本章详细介绍设置边框和边框尺寸的各种属性，像 border、height 和 width 等。

第 7 章　边距与间隙属性。主要介绍如何控制页面元素的边距与间隙，包括 margin 和 padding 复合属性及其子属性。

第 8 章　颜色与背景属性。本章详细介绍各个 CSS 背景属性，像 background、background-color 和 background-position 等。

第 9 章　列表属性。主要对构成列表的符号、图像和位置进行讲解，像 list-style-type、list-style-image 和 list-style 等。

第 10 章　滚动条属性。本章详细介绍如何使用 CSS 的滚动条属性，如设置亮边框、3D 界面边框，设置箭头颜色和拖动区域颜色等。

第 11 章　统揽全局——页面的定位与布局。本章从盒子模型的概念开始，介绍如何对元素进行定位和布局的设计方法。

第 12 章　逐步改善页面的表格。本章主要介绍与表格显示布局有关的属性，如 border-collapse、border-spacing 和 caption-side 等。

第 13 章　逐步改善页面的表单。本章介绍很多方法来增强表单可用性，例如设置访问键

和强调单击等。

第 14 章　滤镜及其他。本章详细介绍各种 CSS 滤镜特效，同时还包含如何设置鼠标指针。

第 15 章　使用 Dreamweaver 提高工作效率。本章主要介绍如何在 Dreamweaver 中创建、保存、修改和编辑 CSS 以提高工作效率，同时还对 Spry 框架有简单的介绍。

第 16 章　与 XML 整合。主要介绍 CSS 在 XML 中的应用，例如嵌入 CSS 和实现隔行变色等。

第 17 章　尽善尽美——合理使用 JavaScript。本章将 CSS 与 JavaScript 结合实现各种页面的绚丽效果。例如变换表格的边框颜色、随着鼠标浮动的文字、会闪的图片等。

第 18 章　CSS 跨浏览器支持技术。本章介绍 4 个典型的 CSS 兼容性问题的解决办法。然后对 CSS Bug 的问题进行了讨论，最后提供了一些有关 CSS 兼容性的建议。

第 19 章　itzcn 书店设计。本章以一个书店为话题，详细讲解如何使用 div+CSS 技术实现各个栏目，如好书推荐、优惠图书和图书分类等。

第 20 章　设计微博首页。本章设计了一个时尚的微博首页，从系统分析、结构分析、设计站点和页面结构，再到各个栏目的实现都遵循 Web 标准，同时包含很多 CSS 技巧。

本书特色

本书中大量内容均来自真实的 Web 网站，力求通过解决实际操作中遇到的问题使读者能够更容易地掌握 CSS 的页面设计。本书难度适中，内容由浅入深，实用性强，覆盖面广，条理清晰。

1) 结构独特

通过“网络教学→基础知识→实例描述→实例应用→运行结果→实例分析”的形式，将每个知识点与实际应用中的问题相结合。

2) 形式新颖

用准确的语言总结概念，用直观的图示演示过程，用详细的注释解释代码，用形象的比方帮助记忆。

3) 技术文档

将一些非常简单的知识点或者理论性的内容安排在这里。通常这些文档没有具体的实际问题，但又是读者必须要了解的，像一些概念和术语。

4) 内容丰富

涵盖了实际使用 CSS 设计时所遇到的 CSS 字体、文本、区块、控制定位、表格和表单样式、特效滤镜、XML 与 CSS 以及跨浏览器兼容性等方面的热点问题。

5) 随书光盘

本书为实例配备了视频教学文件，读者可以通过视频文件更加直观地学习 CSS 的使用知识。

6) 网站技术支持

读者在学习或者工作的过程中，如果遇到难以解决的技术问题，可以直接登录网站 www.itzcn.com 与我们取得联系，作者会在第一时间内给予帮助。

7) 贴心的提示

为了便于读者阅读，全书还穿插着一些技巧、提示等小贴士，体例约定如下。

提示：通常是一些贴心的提醒，使读者能够加深记忆或提供一些建议以及解决问题的方法。

注意：提出学习过程中需要特别注意的一些知识点和内容，以及相关的信息。

技巧：通过简短的文字，指出在应用知识点时的一些小窍门。

读者对象

本书具有知识全面、实例精彩、指导性强的特点，力求以全面的知识性及丰富的实例来指导读者深入地学习 CSS 各方面的知识。既可作为 CSS 初学者的入门教材，也适合 Web 开发人员进一步学习和作为参考。

本书适合以下人员阅读学习：

- Web 应用前端开发者
- Web 应用前端设计者
- 网站开发人员
- 网页制作爱好者
- 其他 Web 前台从业人员
- 各大中专院校的在校学生和相关授课老师

参编人员

本书主要由张芳芳、侯俊杰编写，其他参与编写、资料整理、程序开发的人员还有张浩华、段韶治、李剑龙、崔群法、林田田、王伟平、张水波、张艳梅、胡海静等。

由于编者水平有限，书中难免存在不足和疏漏之处，恳请读者批评指正。

目　录



第 1 章　CSS 快速入门

内容摘要：

CSS 是 Cascading Style Sheets 的缩写，其中文意思是“层叠样式表”。它是一组用于定义 Web 页面外观格式的规则。

使用 CSS 样式设置页面的格式，可以将格式控制代码单独存放，对页面样式进行统一管理。页面内容存放在 HTML 文档中，定义表现形式的 CSS 规则在 HTML 文档的头部或另一个文件中声明。将内容与表现形式的声明相分离，不仅可以增加页面的可维护性，而且还可以使 HTML 文档代码更加简练，HTML 文档结构更加清晰，而且能有效地缩短浏览器的加载时间。

本章我们先了解一下 CSS 的概念、历史和优点，并熟练掌握 CSS 样式表的使用方法以及在 CSS 样式表中添加注释的方法，然后了解一下 CSS 的优先级和度量单位等信息，最后从诸多网页制作工具中挑选一款适合自己的网页制作工具。

学习目标：

- 了解 CSS 的概念、历史和优点
- 掌握几种使用 CSS 样式表的方法
- 掌握对 CSS 样式表添加注释的方法
- 熟悉 CSS 的优先级
- 掌握 CSS 中的度量单位的用法
- 选择一个开发工具

1.1 第一个 CSS 实例

刚接触一个新东西，总会觉得有些手足无措。我刚学 CSS 的时候也是这样，根本就没有这个概念，也不知道从什么地方下手去学习，更不知道从哪个切入点去理解它。

作为本书的第一节，我们就来详细地了解一下 CSS 这个东西，然后通过一个实例来体会一下 CSS 的作用。

**视频教学：**光盘/视频/1/1.1　第一个 CSS 实例.avi　　长度：9 分钟

1.1.1 基础知识——CSS 基础

最早的 Web 服务只是为了共享信息。网络中各计算机之间在互相访问时接收的都是纯文本的内容。

但是随着需求的不断扩大，绚丽的效果也一点点地加入到 Web 页面中。为文本设置丰富多彩的颜色、不一样的字体和字号、不同样式的线条、不同位置不同大小的图片、不同的模块各式各样的背景等。

CSS 自诞生以来，就以其简单的语法、不凡的效果和无与伦比的灵活性，为 Web 的发展做出了不可磨灭的贡献。

1. CSS 基本概念

CSS 全称为 Cascading Style Sheet，可译为“层叠样式表”或“级联样式表”，通常称之为 CSS 样式或样式表。CSS 是一些纯文本内容，存放 CSS 样式表内容的文件扩展名为.css。

CSS 是一种描述性的文本，用于增强或者控制网页的样式，并允许将样式信息与网页内容相分离。

最初，HTML 标签被设计为定义文档结构的功能，通过使用像<h1>、<p>、<table>、<img>等之类的标签，分别在浏览器中展示一个标题、一个段落、一个表格、一个图片等内容。

而页面中内容的布局，由浏览器根据标签表示的内容以从上到下、从左到右的“流”式布局依次排列，如果想要对内容进行定位，则需要使用表格进行分栏控制。

HTML 只是标识页面结构的标记语言。而 Web 发展初期的两大浏览器厂商 Netscape 和 Internet Explorer 为了表现更加丰富的页面效果，争夺 Web 浏览器市场，不断地添加新的标记和属性到 HTML 规范中(比如设置文本样式的 font 元素)，这使得原本结构比较清晰的 HTML 文档变得非常混乱。

而且随着 Web 页面效果的要求越来越多样化，依赖 HTML 的页面表现已经不能满足网页开发者的需求。

CSS 的产生，改变了传统的 HTML 页面的样式效果。CSS 规范代表了 Web 发展史上一个独特的阶段。

2. CSS 的历史

从 20 世纪 90 年代初 HTML 被发明开始，样式就以各种形式存在。不同的浏览器结合它

们各自的样式语言为用户提供页面效果的控制。最初的 HTML 只含有很少的显示属性。

为了满足页面设计者的要求，HTML 逐渐添加了很多显示功能。但是随着这些功能的增加，HTML 变得越来越杂乱，而且 HTML 页面也越来越臃肿。于是 CSS 便随之诞生了。

1994 年哈坤·利提出了 CSS 的最初建议。而正巧当时伯特·波斯(Bert Bos)正在设计一个名为 Argo 的浏览器，于是他们决定一起设计 CSS。

其实当时互联网界已经有过一些统一样式表语言的建议了，但 CSS 是第一个含有“层叠”思想的样式表语言。

在 CSS 中，一个文件的样式可以从其他的样式表中继承下来。读者在有些地方可以使用他自己更喜欢的样式，在其他地方则继承或“层叠”作者的样式。这种层叠的方式使作者和读者都可以灵活地加入自己的设计。

哈坤于 1994 年在芝加哥的一次会议上第一次提出了 CSS 的建议，1995 年他与波斯一起再次提出这个建议。那时候刚刚建立的 W3C 组织对 CSS 的发展很感兴趣，他们为此专门组织了一次讨论会。哈坤、波斯和其他一些人是这个项目的主要技术负责人。1996 年底 CSS 初稿已经完成。同年 12 月 CSS 规范的第一个版本发布。

1997 年初，W3C 组织负责 CSS 的工作组开始讨论第一版中未涉及的问题。其讨论结果组成了 1998 年 5 月发布的 CSS 规范第二版。

从 2001 年开始，CSS 3 提上研究议程。但至今为止，尚未有一个完整的版本发布。

CSS 3 提供了多种新途径去改善你的设计工作，且做了不少重要的变化，值得期待！



因为现在 CSS 3 还没有正式发布，所以本书讲解的所有知识点都是以 CSS 2 为基础的。

3. CSS 的特点

CSS 为 HTML 标记语言提供了一种样式描述，定义了其中元素的显示方式。CSS 在 Web 设计领域是一个突破。我们使用它修改一个小的样式便可更新与之相关的所有页面元素。

总体来说，CSS 具有以下特点。

1) 丰富的样式定义

CSS 提供了丰富的文档样式外观，以及设置文本和背景属性的能力；使用 CSS 可以为任何元素创建边框、设置边框与其他元素间的距离，以及元素边框与元素内容间的距离；还可以随意改变文本的大小写方式、修饰方式以及其他页面效果。

2) 易于使用和修改

CSS 可以将样式定义在 HTML 元素的 style 属性中，也可以将其定义在 HTML 文档的 header 部分，还可以将样式声明在一个专门的 CSS 文件中，供 HTML 页面引用。总之，CSS 样式表可以将所有的样式声明统一存放，统一管理。

另外，我们可以将相同样式的元素进行归类，使用同一个样式进行定义；也可以将某个样式应用到所有同名的 HTML 标签中；也可以将一个 CSS 样式指定到某个页面元素中。

如果要修改样式，我们只需要在样式列表中找到相应的样式声明进行修改即可。

3) 多页面应用

CSS 样式表可以单独存放在一个 CSS 文件中，这样我们就可以在多个页面中使用同一个

CSS 样式表。CSS 样式表理论上不隶属于任何页面文件，在任何页面文件中都可以将其引用。这样就可以实现多个页面风格的统一。

4) 层叠

简单地说，层叠就是对一个元素多次设置同一个样式，这将使用最后一次设置的属性值。

比如我们对一个站点中的多个页面使用了同一套 CSS 样式表，而某些页面中的某些元素想使用其他样式，我们就可以针对这些样式单独定义一个样式表应用到页面中。这些后来定义的样式将对前面的样式设置进行重写，我们在浏览器中看到的将是最后设置的样式效果。

5) 页面压缩

在使用 HTML 定义页面的网站中，往往需要大量或重复的表格和 font 元素形成各种规格的文字样式，这样做的后果就是会产生大量的 HTML 标签，从而使页面文件的大小增加。

而将样式的声明单独放到 CSS 样式表中，可以大大地减小页面的体积，这样在加载页面时使用的时间也会大大地减少。

另外，CSS 样式表的复用更大程度地缩减了页面的体积，减少下载的时间。

4. 使用 CSS 的优势

一项新技术的发展壮大需要其有许多不可替代、不可超越的优点，才能从多种技术中脱颖而出。

CSS 做到了这一点，是因为它有以下优点。

1) Web 页面样式与结构分离

HTML 并不用来控制网页的格式及外观，而是用来控制页面的结构。所以 CSS 样式表单独存放使 Web 页面保持样式和结构相互分离。

2) 页面下载时间更快

CSS 可以重用样式，所以对样式进行设置的代码将会大大减少。另外，因为 CSS 样式表可以单独存放到一个文件中，而像 IE 之类的浏览器的缓存功能会自动判断下载过的资源文件，进行重复使用，缩短下载时间。

3) 节省开发和维护的成本

只需改动很少的几个 CSS 文档，就能够轻松控制具有上千页容量网站的外观。所以网页开发者可以更快、更容易地维护及更新大量的网页。

4) 令人满意的版面样式控制

CSS 控制排版样式功能要比<table>、<font>标签强很多，因此网页设计者不再需要使用<font>标签或单独的图片创建导引线、更改字体颜色、字体大小及种类等。

5) 轻松创建及编辑

创建和编辑 CSS 如同 HTML 一样容易，利用简单的记事本就可以完成。当然也可以使用像 Dreamweaver 一样的 IDE(Integrated Development Environment，集成开发环境)来辅助开发。

6) 兼顾打印和 Web 页面设计

CSS 样式表创建的外观样式能够同时适应浏览器、打印等环境使用。

7) 较好地控制元素在 Web 页面中的位置

CSS 的定位属性允许用户精确地定义元素出现的相对位置，相对于其他元素或浏览器窗体本身。

8) 有利于搜索引擎的搜索

使用 CSS 样式表以后的 HTML 仅仅被用来创建结构，简洁的页面对搜索引擎的友好程度大大增加，可以更加有效地利于搜索引擎收录，所以网页被搜索到的几率就会提高。

1.1.2 实例描述

以 div+CSS 为主流技术的 Web 页面技术现在的发展正如日中天，与 table 相比，div 具有加载速度快、利于搜索引擎优化的特点。

本实例我们就以 div+CSS 技术设计一个简单的个人网站页面。

1.1.3 实例应用

【例 1-1】 第一个 CSS 实例

(1) 在硬盘上新建一个目录，作为我们站点的根目录。

(2) 在站点根目录中创建名为 css 和 imgs 的文件夹，用于存放 CSS 文件和图像资源文件。

(3) 在刚才创建的 imgs 文件夹中存放三个本实例要用到的图像文件，分别是 banner.gif、container_bg.gif 和 header_bg.gif。

(4) 在站点根目录下创建一个文本文件，命名为 tortoise.html。

(5) 用记事本编辑 tortoise.html 文件，添加内容，如下所示：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gbk" />
<title>可爱的小乌龟</title>
<link rel="stylesheet" type="text/css" href="css/tortoise.css"/>
</head>
<body>
<div id="container">
  <div id="header line">
    <div id="header">
      <h1>可爱的小乌龟</h1>
      <ul>
        <li><a href="#">首页</a></li>
        <li><a href="#">我的日记</a></li>
        <li><a href="#">我的照片</a></li>
        <li><a href="#">给我留言</a></li>
      </ul>
    </div>
  </div>
  <div id="body">
    <div id="banner">
      <div id="site_discription">最近买了一只可爱的小乌龟！开心！！！</div>
```

```
    </div>
    <div id="main">
      <div class="log">
        <h3>我的小乌龟</h3>
        <div class="content">
          <p> 小乌龟身穿一身异常坚固的&ldquo;铠甲&rdquo;，伸着四条粗壮的腿，威风凛凛，
活像一个&ldquo;大将军&rdquo;。可这位&ldquo;将军&rdquo;胆子特别小，一旦&ldquo;临阵
&rdquo;就成了名副其实的缩头乌龟了。如果有个人影在它前面晃几下，它也会吓得直剩下龟壳
&mdash;&mdash;&mdash;头、尾、四肢全缩了进去，过了好一会，它才把脑袋偷偷地探出来。</p>
        </div>
        <div class="log_descr">2010-8-3 17:54</div>
      </div>
      <div class="log">
        <h3>小乌龟的新家</h3>
        <div class="content">
          <p>    我知道小乌龟都很喜欢阳光，所以让爸爸给它们做了一间有小楼梯和小阳台的玻璃
房子。看爸爸做好了之后，我就急忙把小乌龟送到了新家中。</p>
          <p>    起先，它在水中自由自在地游泳，好像要游遍自己新房间的每个角落。游累了，它
便停下脚步，把可爱的头抬起并伸出水面，左看一眼，右看一眼，好像处处都有好事、开心事在等着它。
</p>
        </div>
        <div class="log_descr">2010-8-1 10:21</div>
      </div>
    </div>
    <div id="sidebar">
      <h2>站点搜索</h2>
      <input />
      <button type="submit">搜索</button>
      <h2>自我介绍</h2>
      <p>我叫张小山<br />
        今年 11 岁<br />
        现在上小学五年级<br />
        我喜欢画画，也喜欢小动物</p>
    </div>
  </div>
  <div id="footer"> Copyright &copy; 2010 张小山 <a href="#">可爱的小乌龟</a>
</div>
</div>
</body>
</html>
```

(6) 在刚才创建的名为 css 的文件夹中创建一个文本文件，命名为 tortoise.css。

(7) 用记事本编辑 tortoise.css 文件，添加如下内容：

```
@charset "utf-8";

/* -- Global -- */
html,body,div{ margin:0; padding:0; }
body{ font-size:13px; font-family:"宋体", Arial; line-height:21px; }
```

```
/* -- #container -- */
#container{ background-color:#95BA46; padding-bottom:20px;
background-image:url(../imgs/container_bg.gif); background-position:bottom;
background-repeat:repeat-x; }

/* -- #header -- */
#header line{ background-image:url(../imgs/header bg.gif);
background-repeat:repeat-x; }
#header{ height:59px; width:740px; margin:auto; }
#header h1{ margin:0; color:#95BA46; font-size:30px; padding:20px 0 0 0;
float:left; }
#header ul{ list-style:inline; float:right; }
#header ul li{ display:inline; }
#header ul li a{ padding:17px 5px; color:#FFF; font-size:17px;
text-decoration:none; }
#header ul li a:hover,#header ul li a:active{ background-color:#5f6e42;  }

/* -- #body -- */
#body{ background-color:#FFF; padding:15px; width:710px; margin:auto; }
#body:after { content: "."; display: block; clear: both; height: 0; visibility:
hidden }
/* -- #banner -- */
#banner{ width:708px; height:182px; border:solid 1px #333;
background-image:url(../imgs/banner.gif); }
#site_discription{ margin:70px 0 0 30px; font-size:14px;}

/* -- #main -- */
#main{ width:430px; float:left; padding-right:20px; }
.log h3{ border-bottom:solid 1px #95BA46; }
.log .log_descr{ text-align:right; color:#888; }

/* -- #sidebar -- */
#sidebar{ width:240px; float:right; padding-left:20px; }
#sidebar h2{ border-bottom:solid 1px #95BA46; }
/* -- #footer -- */
#footer{ width:738px; font-size:14px; color:#FFF; background-color:#45444D;
text-align:center; margin:auto; border:#fff solid 1px; padding:2px 0; }
#footer a{ color:#FFF; background-color:#45444D; text-decoration:none; }
```

至此所有的编码工作都已完成，下面我们来运行一下。

1.1.4　运行结果

双击运行站点根目录中的 tortoise.html 文件，执行结果如图 1-1 所示。

从上面的代码中可以看到，纯正的 div+CSS 布局的效果非常漂亮，页面结构也很简单。

如果少了 CSS，页面将会是什么样呢？



图 1-1　div+CSS 布局效果

下面我们在 tortoise.html 文件中删掉这一行代码：

```
<link rel="stylesheet" type="text/css" href="css/tortoise.css"/>
```

然后保存，运行该页面，其效果如图 1-2 所示。

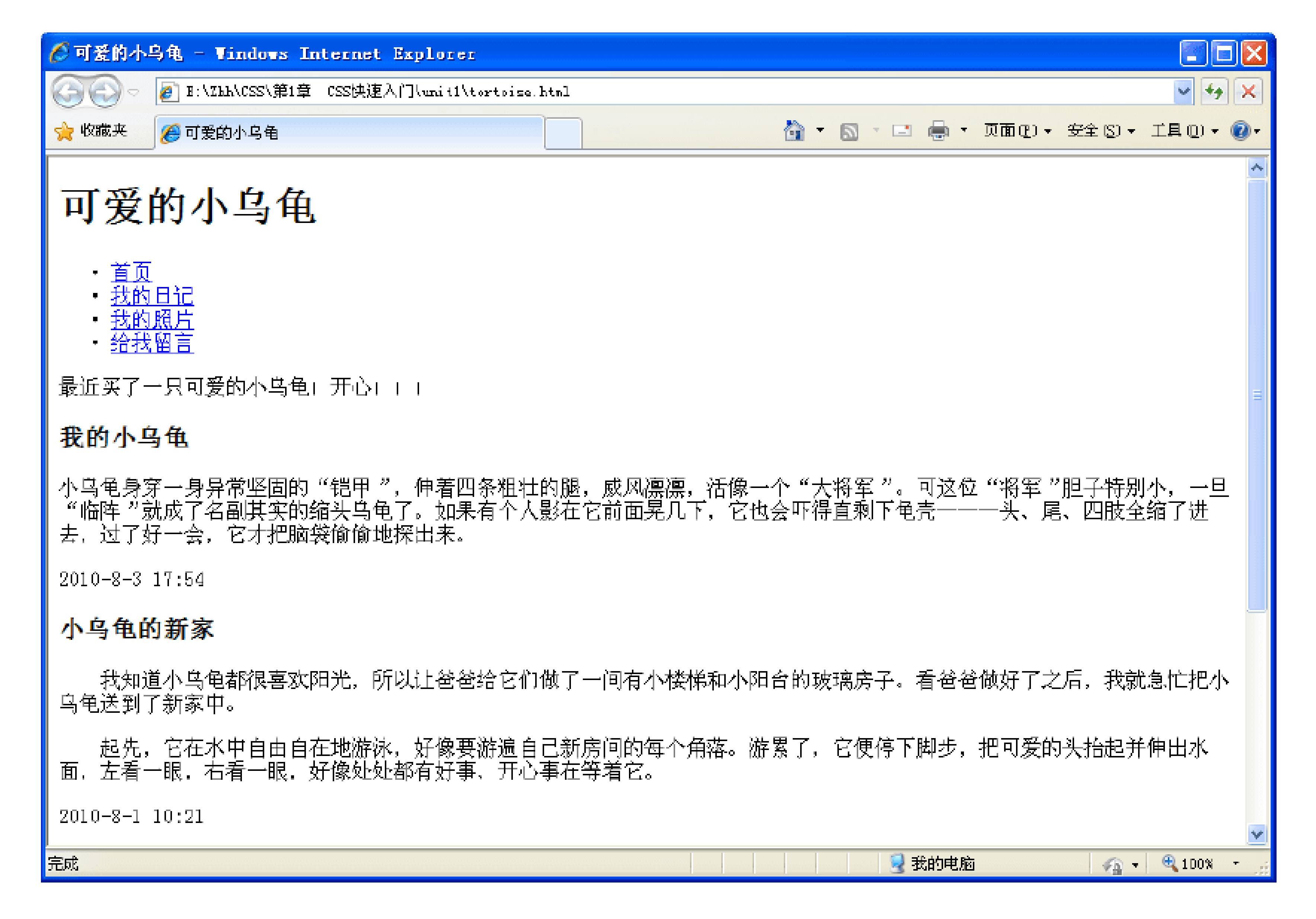


图 1-2　没有 CSS 的 Web 页面

怎么样，是不是觉得惨不忍睹啊。鉴于此，想必大家对 CSS 的作用应该有一个比较深刻的体会了吧。

1.2　使用 CSS 样式

CSS 样式表一个非常强大的特征就是灵活、方便、独立。

传统的 HTML 元素内嵌的样式设置方式使 HTML 代码显得非常杂乱，啰嗦而且不易维护。相对于此，CSS 样式表提供了多种使用样式的方式，让我们可以更加灵活地配置和管理 CSS 样式表。

下面我们就来详细了解一下这些方式。

**视频教学：**光盘/视频/1/1.2　使用 CSS 样式.avi　　长度：9 分钟

1.2.1　基础知识——CSS 样式的使用方式

CSS 样式表是因为 Web 而存在的，所以 CSS 样式终归是要应用到 HTML 页面中的。

与传统的 HTML 元素内嵌的样式设置方式相比，CSS 样式表的一个非常重要的特点就是灵活。它支持将 CSS 样式作为属性设置到页面元素中；也支持将 CSS 样式表集中声明在页面头部；还支持将所有的 CSS 样式表内容存放到一个单独的文件中，在 HTML 页面中引入。

总体来说，CSS 提供了以下 4 种使用 CSS 样式表的方式：内嵌样式、内部样式表、导入样式表和链入样式表。下面我们对它们分别进行介绍。

1. 内嵌样式

内嵌样式表与传统的 HTML 元素内嵌样式的方式非常相似，也是直接将样式定义代码作为属性定义在 HTML 元素中。这种方式的特点是方便、直观，但是因为其与 HTML 页面结构的强耦合，使得样式的调整与页面结构无法分离，因此给维护人员带来非常大的不便。

但是，这种方式的方便性使得我们在设计非常简单的页面代码时更加直观，也使整个页面显得不那么复杂，所以 CSS 没有把它抛弃掉。

使用内嵌 CSS 样式也非常简单。我们可以直接将 CSS 样式声明的代码作为一个字符串设置到 HTML 元素的 style 属性当中即可。

例如，要在页面中显示一行颜色为红色的、字体大小为 13 像素的文本，可以这样做：

```
<span style="color:red; font-size:13px;">这是一行文本</span>
```

同样，对应的传统的 HTML 内嵌样式代码如下：

```
<font color="red" size="2">这是一行文本</font>
```



font 标签里的 size 属性可能会因为不同的解析方式会有不同的差异(比如 Dreamweaver 的视图方式就不一样)，不过现在主流的浏览器(Internet Explorer 8、Firefox、Safari、Chrome)都能正常显示。

2. 内部样式表

内部样式表是将所有的样式声明代码放置于页面头部。你可以在页面头部的 head 元素内部添加一个 style 元素，用于存放样式表配置。



其实 style 元素和 script 元素一样，可以放置在页面中的任何位置，所以没有必要一定放在 head 元素内。但是，按 W3C 标准的要求是需要将其放在 head 元素内统一管理，而且放置在其他地方也没什么特别的好处，所以建议放在 head 元素中。

style 元素是使用一个开始标签<style>和一个结束标签</style>括起来的一个代码段。元素的内容就是样式表的所有配置信息。比如我们要为页面中所有的 div 设置一个红色的 1 像素宽的实线边框，可以使用以下代码实现：

```
<style type="text/css">
div {
    border:solid 1px #c00;
}
</style>
```

3. 导入样式表

如果将 CSS 样式表声明的代码单独放在一个扩展名为.css 的文本文件中，就可以在页面中的 style 元素里使用@import 语句将外部的 CSS 样式表文件导入。

例如上面为 div 设置的边框样式，可以使用样式表单独存放样式的设置信息。

可以先新建一个名为 test.css 的样式表文件，内容如下：

```
div {
    border:solid 1px #c00;
}
```

然后在页面中的 style 元素中使用@import 将该文件导入，代码如下：

```
<style type="text/css">
@import "test.css";
</style>
```

这样我们就可以正常地使用 CSS 样式表文件中定义的样式了。

4. 链入样式表

链入样式表的方式和导入样式表的方式差不多。它们都是将样式单独存放，然后在 HTML 页面中引入该样式表文件。

链入样式表是在 head 元素中使用 link 元素将其引入，外部样式表文件的名称作为 link 元素的 href 属性的值设置即可。

比如刚才创建的 CSS 样式表文件 test.css 就可以使用链入样式表的方式引入 HTML 页面中使用，代码如下：

```
<link rel="stylesheet" type="text/css" href="test.css"/>
```

那么导入样式表和链入样式表有什么区别呢？

link 元素属于 HTML 级的，而@import 属于 CSS 级的。在页面运行的时候，浏览器获得 HTML 代码先进行 HTML 标签的分析，HTML 代码在遇到 link 标签时会去加载 link 标签所链接的资源文件；而执行到包含@import 语句的 style 元素时，浏览器只会认为它是一个 HTML 元素，并不会去顾及这个元素的内容。而在执行 HTML 文档的时候，浏览器会直接执行 link 元素链入的样式表内容；到执行 style 元素的时候，才发现需要导入外部 CSS 样式表文件，这时浏览器才会异步请求 CSS 样式表文件，并同时向下执行 HTML 文档的其他代码。

这样在使用@import 的时候，在网络比较慢的时候加载页面样式经常会有些延迟，所以一般页面开始的样式非常混乱难看，过一会儿才会恢复本来的样子。

1.2.2　实例描述

使用 CSS 样式非常简单。前面我们已经讲过 4 种 CSS 样式的使用方法，下面这个实例，我们就使用这 4 种方法配合着做一个简单的页面。

1.2.3　实例应用

【例 1-2】使用 CSS 样式

(1) 先选择一个目录，用于作为站点的根目录。
(2) 在目录下新建两个文件夹，命名为 css 和 imgs。
(3) 在 imgs 文件夹中放入一个作为页面背景的图像文件，名为 ypz.jpg。
(4) 在 css 文件夹中添加两个 CSS 样式表文件，分别命名为 ypz1.css 和 ypz2.css。
(5) 在 ypz1.css 中添加一段设置页面背景的样式代码，如下所示：

```
@charset "utf-8";

body {
    background-image:url(../imgs/ypz.jpg);
    background-position:center top;
    background-repeat:no-repeat;
}
```

(6) 在 ypz2.css 中添加一段设置段落字号的样式代码，如下所示：

```
@charset "utf-8";

p {
    font-size:21px;
}
```

(7) 接下来在站点根目录下创建一个名为 QuietNightThink.html 的文件。
(8) 使用 Windows 记事本打开并修改 QuietNightThink.html 文件，添加内容如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Quiet night think</title>
<link rel="stylesheet" type="text/css" href="css/ypz1.css"/>
<style type="text/css">
@import "css/ypz2.css";
body {
    text-align:center;
    padding-top:70px;
    color:#805231;
    font-family:"Monotype Corsiva";
}
</style>
</head>

<body>

<h1 style="font-size:30px; padding-bottom:20px;">Quiet night think</h1>

<p>Bed bright moonlight,</p>
<p>Doubt is the ground frost.</p>
<p>Withdrawing my eyes bright moon,</p>
<p>Bow think hometown.</p>

</body>
</html>
```

页面中的代码用到了前面讲的 4 种 CSS 样式用法。接下来我们运行一下看看效果。

1.2.4 运行结果

使用 Internet Explorer 打开 QuietNightThink.html 文件，运行结果如图 1-3 所示。

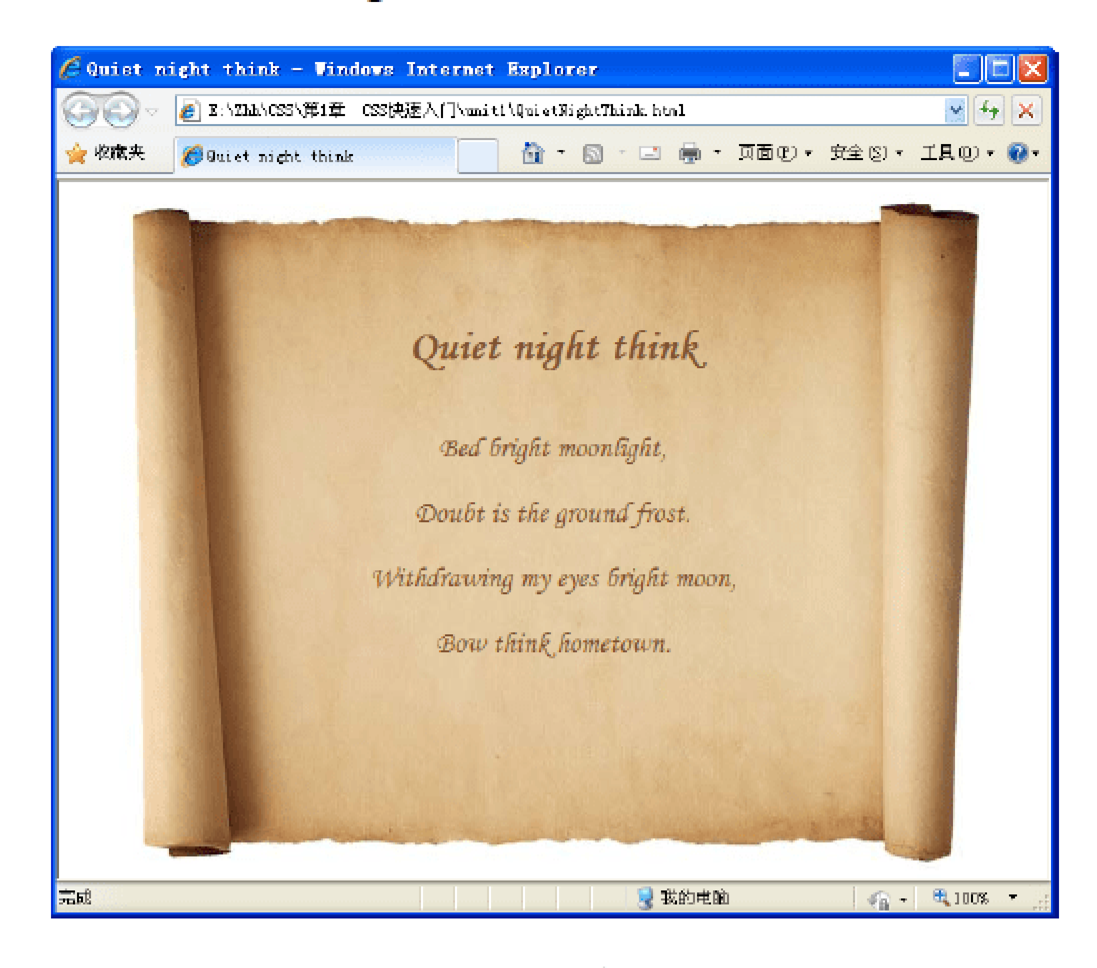


图 1-3 运行结果

1.2.5 实例分析



源码解析：

本实例使用元素内嵌的样式设置了标题元素 h1 的字体和下间距属性；在页面 head 元素内使用内部样式表设置了页面的对齐方式、顶部间距、字体颜色、字体样式等属性；使用 link 元素链入了一个外部 CSS 样式表 ypz1.css，设置了页面的背景图片等属性；在页面 style 元素中使用@import 语句导入一个外部 CSS 样式表 ypz2.css。

1.3 CSS 的语法

CSS 也是一种语言，一种标识页面样式的标记语言，所以它也有自己的语法。

在 CSS 样式代码中，一条 CSS 样式声明的语句可以由两部分组成：选择符和样式列表。当然，这里要将使用元素内嵌的样式排除在外。

选择符是定义使用该样式列表的 HTML 元素的标识符，它可以是一个类(HTML 元素的 class 属性)、一个元素 id 或者 HTML 元素的标记名称。

样式列表是一个集合，这里使用一对花括号来表示一组样式。一个样式的设置是由一个样式属性名称和一个属性值组成的，它们之间用冒号隔开。一个样式集合中多个样式之间使用分号分隔。

CSS 样式表定义语句的格式如下：

```
selector {
    property:value;
    property:value
}
```

就比如 1.2 节实例中用过的对 body 元素设置的样式：

```
body {
    text-align:center;
    padding-top:70px;
    color:#805231;
    font-family:"Monotype Corsiva";
}
```

1.4 增强 CSS 的可读性——使用注释

CSS 毕竟不是我们通常使用的自然语言，所以经常会有些令人难懂，或者有看起来不是很直观的地方。这样我们在以后维护的时候就需要浪费很多时间去阅读理解这些设置代码，非常不方便。

当然，使用自然语言对 CSS 样式表进行定义在当前情况下显然是不可能的事情，所以我们需要想一些办法，提高 CSS 样式代码的可读性。

对于这个问题，有一个首选的方式就是使用注释。

注释是对代码的一种说明性的标记，可以使用它对程序代码进行说明，使其更加易读易懂。

在 CSS 中可以使用斜杠“/”和星号“*”的组合来标记 CSS 文件的注释。比如下面代码就定义了一行 CSS 注释：

```
/* 这是一行注释 */
```

当然，前面章节中定义的 CSS 样式也可以更进一步地进行优化，如下：

```
body {  /* 对 body 标记添加样式 */
    text-align:center;  /* 设置元素中文本对齐方式为 center(居中) */
    padding-top:70px;  /* 设置元素顶部填充空间为 70px(像素) */
    color:#805231;  /* 设置元素的文本颜色为#805231 */
    font-family:"Monotype Corsiva";  /* 设置元素中文本的字体为 Monotype Corsiva */
}
```

这样是不是就会觉得更加清晰易读了？

当然，有些常用的，一看就明白的属性没必要加注释，比如 color 等。这些没必要添加注释的代码省去注释会显得非常简洁，所以注释也不是越多越好。具体使用方式视情况而定，灵活运用即可。

1.5 CSS 优先级

所谓 CSS 优先级，即是指 CSS 样式在浏览器中被解析的先后顺序。作为页面设计人员，了解这一点是十分重要的，因为这直接关系到页面最终展示的效果。

要了解 CSS 样式的优先级，我们必须知道 CSS 样式规则的一个很重要的特性——继承性。

 **视频教学：**光盘/视频/1/1.5 CSS 优先级.avi　　**长度：8 分钟**

1.5.1 CSS 的后来居上原则

由于 CSS 只是一组规则，因此当规则重复定义时并不会有什么警告或者提示。因为，有时候我们需要对样式进行重新定义而编写重复的规则。

比如我们在页面中导入多个样式表文件，多个样式表文件中都对 body 元素的背景进行了不同的设置，系统则自动使用最后一次匹配的样式进行展示。

举一个简单的例子，比如我们不小心在同一个样式规则中多次设置 div 元素中文本的颜色，代码如下：

```
div {
    color:red;
    color:blue;
}
```

上面代码先后两次为 div 标记设置了文本颜色属性，页面中展示出来的将是最后一次设置的值(这里是蓝色(blue))。

另外，不同 CSS 样式的引入方式也可能会对该原则造成影响，这一点主要体现在引入外部 CSS 样式表文件中。

引入外部 CSS 样式表文件有两种方式，使用起来各有差异，详细说明如下。

- @import 语句导入：该语句导入的 CSS 样式表文件内容会自动加入到当前 style 元素的顶端执行，也就是说导入的样式先于当前 style 元素中配置的任何 CSS 样式执行。如果当前 style 元素中导入多个 CSS 样式表文件，那么它们之间则按导入的先后顺序进行解释执行。当然多个 style 元素互相之间就没有什么影响了。
- link 元素链入：该元素的使用比较简单，它只在使用 link 元素链入样式表文件的位置进行解析，先后顺序以 link 标签的顺序为准。这一点没有什么异议。

当然，如果上升到 HTML 文档级别，style 元素和 link 元素执行的先后顺序就以它们在 HTML 文档中的先后顺序为准。

1.5.2　CSS 的继承性

继承是 CSS 一个非常重要的特性，它是依赖于 HTML 中元素节点的父-子级关系(元素的包含关系)的。它允许样式不仅可以应用于当前指定的元素，还可以应用到当前元素下的子元素(当然也包括“孙子”级以上的元素)中。

比如页面中有以下代码：

```
<div id="recolor">
    当前元素的内容
    <div>
        子元素中的内容
        <div>
            下一级子节点(孙子级节点)中的内容
        </div>
    </div>
</div>
```

然后为其添加样式：

```
<style type="text/css">
#recolor {
    color:red;
}
</style>
```

这样页面中的所有文本的样式都被设置为红色(red)了。

当然，也不是所有的 CSS 样式都能够被继承。

CSS 提供了智能分析的功能，一些我们一定不想被继承的样式，则不会被子元素继承，比如 border、margin、padding、background 等。因为一旦这些样式被继承，则会对页面的样式产生非常大的影响，这些是我们不想看到的。

比如刚才的页面代码，我们为其添加一个边框样式，代码如下：

```
<style type="text/css">
#recolor {
    color:red;
    border:solid 3px blue;
}
</style>
```

这样如果 border 样式被继承，页面中每一行文字将会被一个蓝色的边框套住，但事实上并不是这样，如图 1-4 所示。

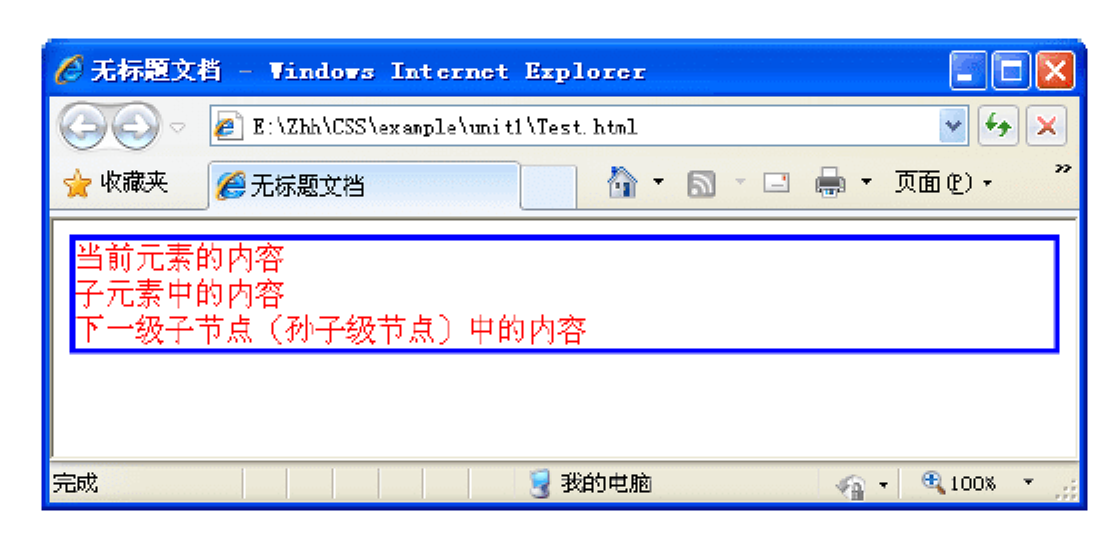


图 1-4　样式继承运行结果

从图 1-4 中可以看到，虽然所有的文本都变成了红色，但是并不是所有的 div 元素都添加了边框，只是最外层的 div(也就是样式设置的当前元素)被添加了一个 3 像素宽的蓝色实线边框。

1.5.3　CSS 的选择器优先级

前面提过“后来居上原则”和“继承性”，还有一个虽然不很引人注意，但是非常重要的优先级规则就是选择器匹配的优先级，这一点是 CSS 优先级中最复杂、最不容易理解的。

一个选择器中可以有多个选择符。CSS 中的选择符主要分为三类：id 选择符、class 选择符和 HTML 标签选择符。

显然，id 选择符是为了根据 id 选择匹配的 HTML 元素，class 选择符是根据元素的 class 属性进行匹配，HTML 标签选择符就是根据元素的标记进行匹配的。



这里简单提一下选择符，具体用法在以后的章节中会进行详细讲解。

在一条 CSS 样式规则的选择器中，可能会有多个选择符组合进行元素的精确定位。比如下面的样式定义规则：

```
#bbb div {
    font-size:30px;
}
```

上面代码定义了一个选择器“#bbb div”，其中包含两个选择符“#bbb”和“div”。

选择器优先级就是根据选择符的权重进行组合并选择使用元素的样式。上面说的三类选择

符中，id 的权重最高，class 次之，HTML 标签的权重最低。

在 CSS 样式表中匹配的规则是查询选择器中每一类选择符的个数，然后进行分类。组成一组三个整数，按选择符的权重排序，进行比较。

比如上面的选择器中使用了一个 id 选择符和一个标签选择符，那么上面选择器的权重就是“1、0、1”(id、class、HTML 标签)。

比较的时候先比较 id 选择符对应的值，哪个样式规则选择器中的值大，就取哪个规则中的值；接下来比较 class 选择符对应的值；然后比较 HTML 标签选择符的值。

如果说两个 CSS 样式选择器的权重相同，则按“后来居上”的原则匹配。

比如页面中有这样一段 HTML 代码：

```
<div id="bbb" class="xxx">
    <div id="ddd" class="ccc">这是一行文本</div>
</div>
```

而且在页面头部的 style 标签中声明了以下样式声明代码：

```
.xxx #ddd {
    font-size:10px;
}
#bbb .ccc {
    font-size:30px;
}
```

因为两个样式规则的选择器权重比相同，所以使用后面的那个样式规则，页面中文本大小将为 30 像素。

不过需要注意一点，并不是所有优先级低的样式规则都不起作用。样式优先级的权重比只是在样式设置重复或者冲突的情况下才起作用。

可能这句话不好理解，我们来举个例子。比如上面的样式声明做这样的修改：

```
.xxx #ddd {
    font-size:10px;
    color:Green;
}
#bbb .ccc {
    font-size:30px;
}
```

这样虽然页面中字体大小是 30px，但是其同样使用了优先级低的样式规则中的修改字符颜色的样式。页面中将显示一行绿色的 30px 大小的文字，如图 1-5 所示。

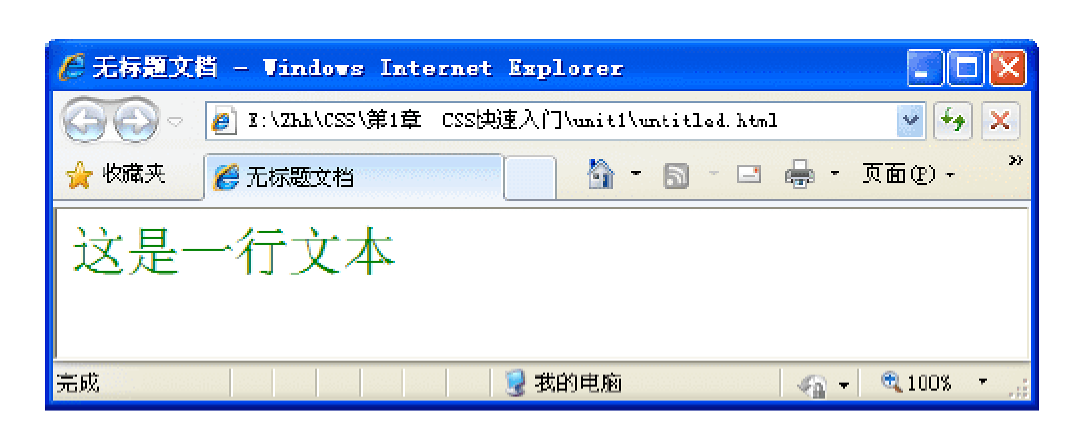


图 1-5　样式选择器的优先级

1.5.4 !important 声明强制优先

CSS 优先级中还有一个最无敌的声明，就是!important。

在 CSS 样式表中，带有!important 声明的样式优先使用，它的优先级会超越任何地方、任何方式的样式声明，包括行内 style 属性中声明的样式。

比如下面代码：

```
<style type="text/css">
div{ color:red !important; }
</style>
<div style="color: blue; ">这是一行文字</div>
```

这里的这行文字在浏览器中仍然显示为红色字体。

所以，CSS 样式匹配的优先权遵循以下顺序：

(1) !important 声明高于一切。

(2) 内嵌样式次之。

(3) 样式表中声明按选择器的权重排序。

(4) 最后按浏览器执行样式表的顺序，遵循“后来居上”原则匹配样式。

当然最后还要强调一点：如果没有设置样式的，则自动继承父级节点的样式。

1.6 CSS 中风格不一的文字

CSS 既然要标识样式，就少不了使用数量词的形式表示物体的度量。当然，使用数量词就少不了单位。

平常我们生活中经常使用的单位有长度、重量、温度、时间、速度、面积、体积等很多种。而在 Web 页面中，我们只需要使用 CSS 样式在页面中展示一个平面图像即可，所以这里我们只需要表示一些长度、颜色之类的可视化信息。

下面我们就来了解一下 CSS 中的单位。

**视频教学：**光盘/视频/1/1.6 CSS 中风格不一的文字.avi　　长度：8 分钟

1.6.1 基础知识——CSS 中的单位

CSS 中的值一般可以分为几种：数值、字符串、常量、url 等。

数值在 CSS 中一般指的是长度，是对页面中对象大小的说明；字符串一般是一些特定的值，例如字体等；常量一般指的就是一些特定的值了，比如颜色 red、blue 等；url 就是指一个资源路径了。

其中最常见的也就是长度、颜色、url 这几种单位了，另外还有一种比较特殊的长度单位就是百分比单位。

下面我们分别对它们进行介绍。

1. 长度单位

长度单位主要分为两种：相对长度单位和绝对长度单位。

相对长度单位主要有以下几种。

- em：元素的字体的高度。
- ex：相对于字母 x 的高度。
- px：像素(Pixels)。该单位是相对于显示器分辨率的长度单位。因为计算机可视化界面都主要显示在显示器中，所以 px 是计算机中最常用的长度单位。同样在 Web 页面中，px 也是最常用的单位。

我们在页面中定义元素的宽度和高度经常会使用 px 作单位，例如下面代码：

```
div {
    width:350px;
    height:80px;
}
```



有一个百分比长度单位，也算是一个相对单位。因为其比较特殊，我们将会单独对它进行讲解。

绝对长度单位有以下几种。

- in：英寸 Inches。1 英寸相当于 25.4 毫米。
- cm：厘米(Centimeters)。
- mm：毫米(Millimeters)。
- pt：点(Points)。1 点相当于 1/72 英寸。
- pc：皮卡(Picas)。这个是打印机使用的字体大小。

绝对长度单位视输出介质而定，所以在 Web 页面中使用起来很不方便。



绝对单位的换算关系：1in = 6pc = 72pt = 25.4mm = 2.54cm

2. 百分比单位

百分比单位是一个比较特殊的长度单位。一般它指定的是当前元素本身相对于父级元素的大小。

比如我们经常会在表格的同一行的单元格中使用该单位，以便使同一行的单元格能等比地均分表格的宽度。代码如下：

```
<table style="width:100px;">
<tr>
    <td style="width:30%;"> </td>
    <td style="width:70%;"> </td>
</tr>
</table>
```

这样同一行中的两个单元格一个宽度为 30%，一个宽度为 70%。那么如果表格宽度为 100

像素，它们两个的宽度就分别为 30 像素和 70 像素；如果表格宽度为 300 像素，它们两个的宽度分别为 90 像素和 210 像素。

3. 颜色单位

颜色值有两种形式：使用关键字表示的常用颜色和使用 RGB 值表示的具体颜色值。

颜色关键字比较直观，比如代表红色的 red、代表蓝色的 blue、代表黄色的 yellow 和代表绿色的 green。但是使用颜色关键字只能表示最常用的一些颜色，如果需要表示得更加丰富多彩，就可以使用 RGB 值的方式来表示颜色。

用 RGB 值表示颜色有 4 种方式。

- #rrggbb：三个两位的十六进制无符号整数，取值范围在 00～ff 之间，如#e7cc5b。
- #rgb：三个一位的十六进制无符号整数，取值范围在 0～f 之间，如#ce8，等同于第一种形式的#ccee88。
- rgb(r,g,b) ：括号中的 r、g、b 都是一个 0～255 的十进制整数。
- rgb(r%,g%,b%)：括号中的 r、g、b 都是一个 0～100 的十进制数字。

比如要表示蓝色，我们可以使用下面 5 种方式中的任意一种：

```
blue
rgb(0%,0%,100%)
rgb(0,0,255)
#00f
#0000ff
```

4. url

url 是一个地址。

一个 URL 值的格式为

```
url(foo)
```

上面参数 foo 是一个 URL(统一资源定位符)，是一个资源的地址。该地址可以是绝对路径，也可以是相对路径。URL 可以选择用单引号(')或者双引号(")引用，或者直接写进括号也可以。

例如要表示当前目录中的一个名为 Logo.gif 的图片资源，可以使用以下语句：

```
url(Logo.gif)
```

1.6.2 实例描述

CSS 中的单位使用十分频繁，以后大家会大量用到它们。所以本实例只是简单地使用上面讲过的单位在页面中显示一些内容。

1.6.3 实例应用

【**例 1-3**】CSS 中风格不一的文字

(1) 选择一个目录作为站点的根目录。

(2) 在根目录中添加一个 css 文件夹和一个 img 文件夹。

(3) 在 img 文件夹下存放一个用于做背景的图片文件，名为 fgb.jpg。

(4) 在 css 文件夹下创建一个名为 font.css 的 CSS 样式表文件。

(5) 修改 font.css 文件，添加样式声明代码，如下所示：

```
@charset "utf-8";
#px {
    font-size:13px;
}
#ex {
    font-size:3ex;
}
#em {
    font-size:1em;
}
#in {
    font-size:0.3in;
}
#cm {
    font-size:1cm;
}
#mm {
    font-size:6mm;
}
#pt {
    font-size:20pt;
}
#pc {
    font-size:3pc;
}
body {
    background-image:url(../imgs/fgb.jpg);
}
div {
    color:#999;
}
p {
    color:rgb(10%,10%,10%);
}
```

上面代码分别设置了不同 id 使用的不同单位显示的文本。并为页面中 body 元素添加一个背景图片，然后对 div 元素和 p 元素设置文本颜色。

(6) 在站点根目录下添加一个 HTML 文件，名为 font.html。

(7) 使用 Windows 记事本打开并修改 font.html 文件，内容如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="css/font.css"/>
<title>Font</title>
</head>

<body>
<p id="px">天净沙</p>
<p id="ex">秋思</p>
<p id="em">(马致远)</p>
<div id="in">枯藤老树昏鸦，</div>
<div id="cm">小桥流水人家，</div>
<div id="mm">古道西风瘦马。</div>
<div id="pt">夕阳西下，</div>
<div id="pc">断肠人在天涯。</div>
</body>
</html>
```

1.6.4　运行结果

使用 IE 浏览器打开 font.html 文件，运行结果如图 1-6 所示。

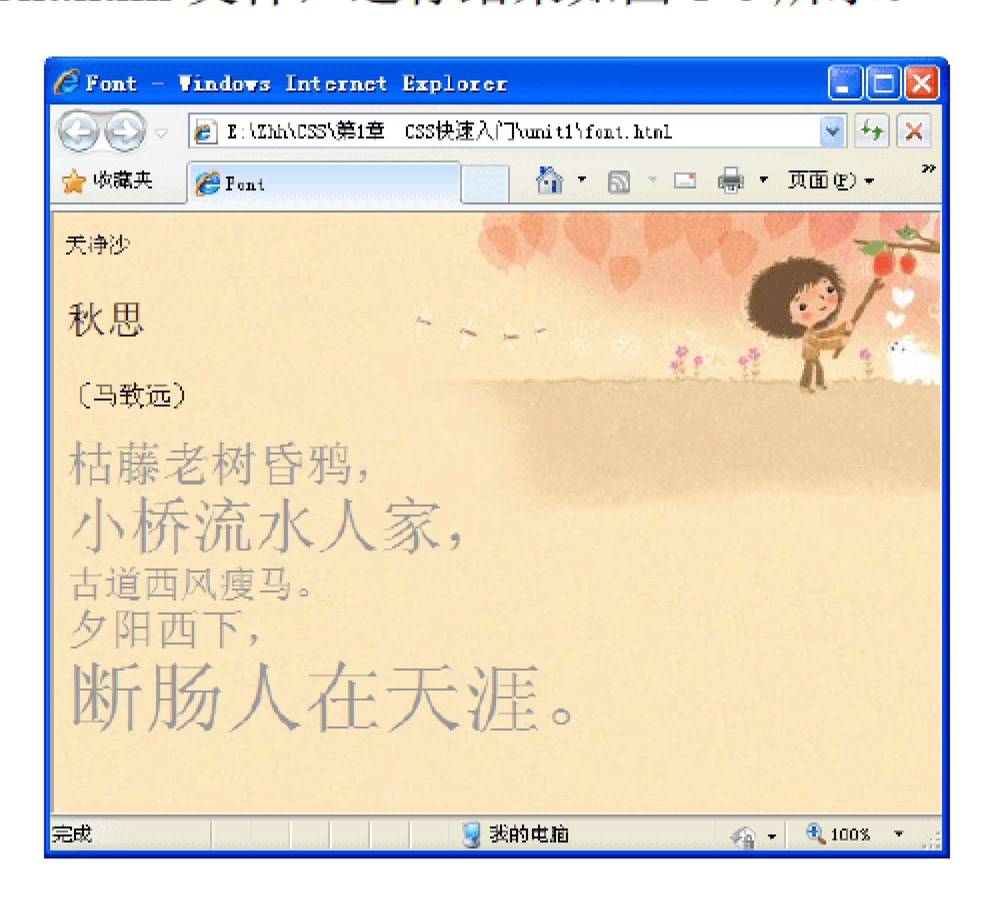


图 1-6　运行结果

1.6.5　实例分析



源码解析：

本实例分别使用不同的长度单位为页面中不同的元素定义文本大小；然后使用两种颜色声明方式为页面中两种 HTML 标签指定浅灰和深灰两种不同的文本颜色；最后使用 url 引用一个图片资源文件为页面中的 body 元素添加一个背景图像。

1.7　选择一款网页制作工具

“工欲善其事，必先利其器。”

尽管选择什么样的网页制作工具不会影响设计的网页质量的好坏，但是如果有一款功能强大，使用简单的网页制作工具往往会让我们的开发工作事半功倍。

例如，前面我们一直使用的是 Windows 系统自带的记事本工具编辑 HTML 文件，没有语法检查、没有代码格式化、没有自动提示，代码编写起来相当痛苦，如图 1-7 所示。

```
QuietNightThink - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Quiet night think</title>
<link rel="stylesheet" type="text/css" href="css/ypz1.css"/>
<style type="text/css">
@import "css/ypz2.css";
body {
        text-align:center;
        padding-top:70px;
        color:#805231;
        font-family:"Monotype Corsiva";
}
</style>
</head>

<body>

<h1 style="font-size:30px; padding-bottom:20px;">Quiet night think</h1>

<p>Bed bright moonlight,</p>
<p>Doubt is the ground frost.</p>
<p>Withdrawing my eyes bright moon,</p>
<p>Bow think hometown.</p>

</body>
</html>
```

图 1-7　记事本编辑 HTML 文档

可以制作网页的工具比较多。因为网页是文本文件，所以只要能处理文字的工具都可以对网页进行编辑，比如前面所说的记事本，或者说一些稍微高级一点的工具，比如 UltraEdit、EditPlus 等。

当然，在 Web 应用程序如此盛行的年代，专业的网页制作工具也层出不穷。其中就有微软公司的 FrontPage。

FrontPage 是作为 Office 办公套件中的组件一起推出的，所以其主界面与其他的 Office 工具十分相似。当然，功能也很强大，使用起来也非常简单。

FrontPage 的主界面如图 1-8 所示。

目前大多数网页制作人员都喜欢使用所见即所得的编辑工具，而 FrontPage 就属于这一类型的。

它出自于软件帝国微软的旗下，优点自然不必多说。

不过，下面我们要请出一位网页制作界的重量级选手——Dreamweaver，它在网页制作方面相对于 FrontPage 来说有过之而无不及。

Dreamweaver 是 Macromedia 公司的产品，它与 Flash、Fireworks 并称网页三剑客，是网页制作工具界的霸主。多少年来独领风骚，连 FrontPage 都不是它的对手。

Dreamweaver 是集网页制作和管理网站于一身的所见即所得的网页编辑器，它是第一套针对专业网页设计师特别开发的视觉化网页开发工具，利用它可以轻而易举地制作出充满动感的

网页。

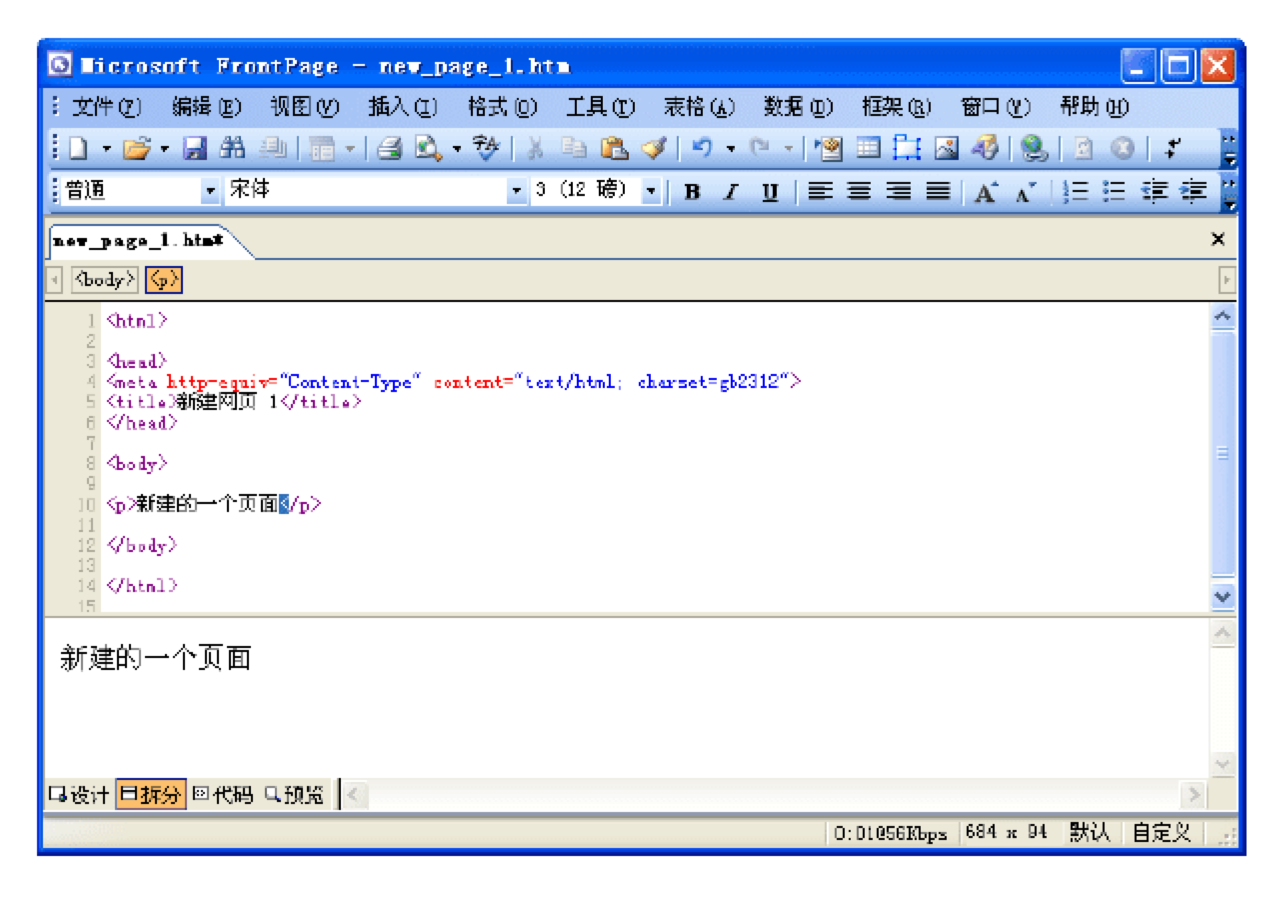


图 1-8　FrontPage 2003 的主界面

Dreamweaver 的主界面如图 1-9 所示。

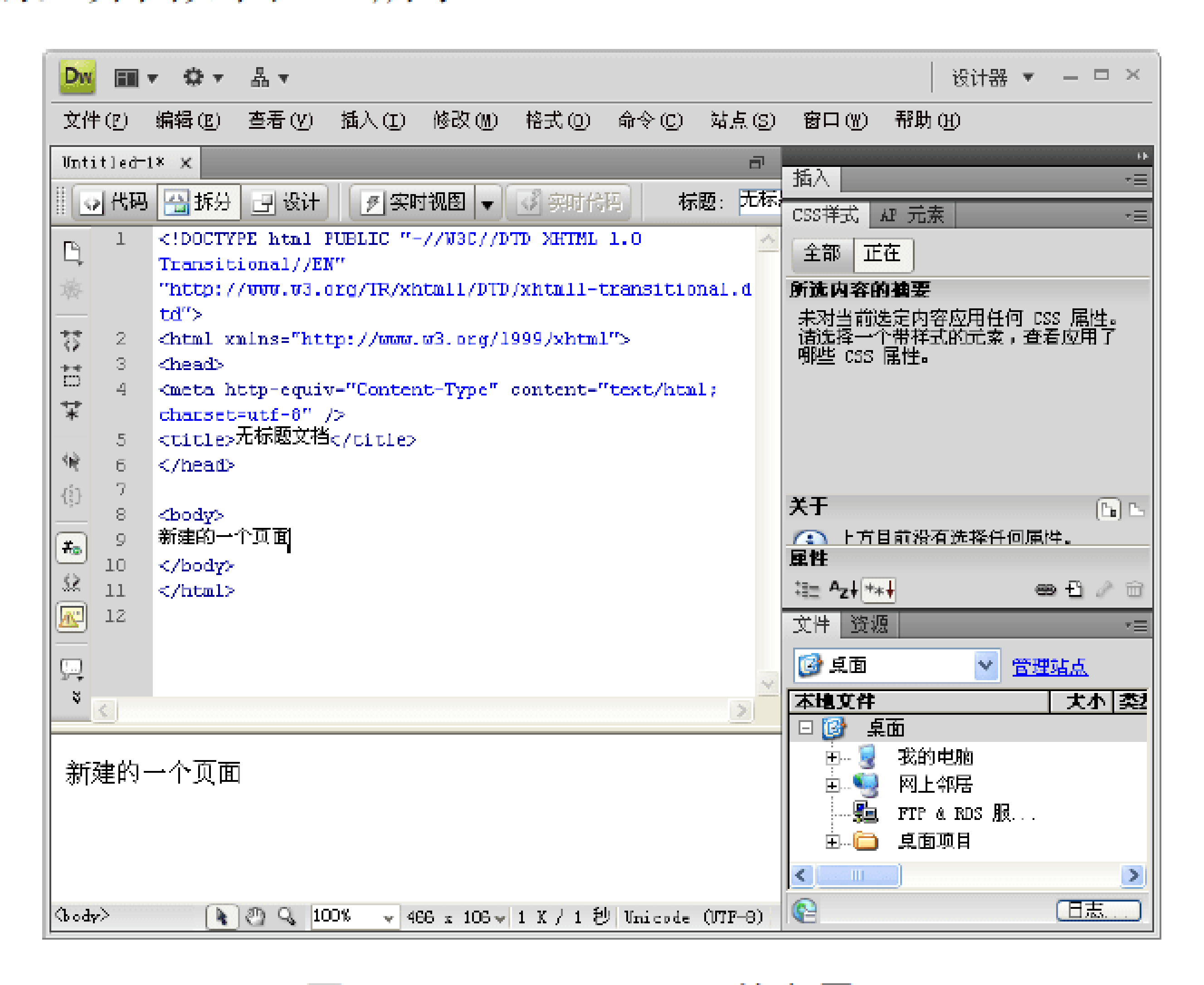


图 1-9　Dreamweaver 的主界面

Dreamweaver 主要有以下优点。

- 最佳的制作效率：Dreamweaver 可以使用鼠标拖拽的方式快速地放入页面元素，它的智能提示的功能可使你更加惬意地编写代码。
- 网站管理：Dreamweaver 使用网站地图可以很方便地管理站点资源和文件。
- 无可比拟的控制能力：Dreamweaver 是唯一提供 Roundtrip HTML、视觉化编辑与原始码编辑同步的设计工具。
- 集成动态开发语言：Dreamweaver 还集成了动态程序开发语言，完全支持 ASP、.NET、PHP、JavaScript 的基本语言和连接操作数据库的功能。

总之，Dreamweaver 是一个非常优秀的网页制作工具。正如它的名字一样，能使你的网页设计工作像编织一个美丽的梦一样美妙。



在本书中，我们将主要使用 Dreamweaver 工具进行讲解和演示。

1.8　常见问题解答

1.8.1　学习 CSS 需要哪些特殊的软件



学习 CSS 需要哪些特殊的软件吗？
网络课堂：http://bbs.itzcn.com/thread-11367-1-1.html

请问学习 CSS 需要哪些特殊的软件或特殊的工具吗？

【解决办法】不用。

CSS 和 HTML 一样，都是纯文本内容，所以在学习 CSS 的时候只需要使用 Windows 自带的记事本(Notepad.exe)就可以了。

不过在实际制作网页的时候可能需要用到 Dreamweaver 等专业的网页设计软件，它们将使得为网页添加 CSS 的工作变得异常简单。

1.8.2　CSS+div 菜鸟级问题



CSS+div 菜鸟级问题。
网络课堂：http://bbs.itzcn.com/thread-11368-1-1.html

我在做网站的时候，有一个问题：全站用 CSS+div 布局，现在如果换分辨率或者换显示器，会有不同的显示效果，我的要求只有一个，那就是不论是换什么显示器，什么分辨率下面看到的结果都是一样的，请问高手们这个问题该如何解决，希望能够讲得详细一些，谢谢。

【解决办法】设定顶级 box 宽度，且为具体的宽度，如 px、em、pt 等，而不是使用百分比控制宽度%。

而且还要注意，你使用的是 div，div 在默认情况下宽度是 100%。

1.8.3　学习 CSS 都需要哪些基础知识



学习 CSS 都需要哪些基础知识。
网络课堂：http://bbs.itzcn.com/thread-11369-1-1.html

听网上说 CSS 不错，我想学 CSS，但对这些东西不了解，我都要学些什么东西啊？

【解决办法】CSS 是对 Web 页面中的内容进行样式设置的，所以学习 CSS 之前你最起码要掌握一些 HTML 的知识。

1.9 习　　题

1. 填空题

(1) CSS 是__________的缩写形式。

(2) 有以下代码:

```
<style type="text/css">
div{ color:blue; /* 蓝色 */ }
#d1{ color:red; /* 红色 */ }
#d2{ color:yellow; /* 黄色 */ }
</style>

<div id="d1">
  <div id="d2">
    <div id="d3"> 请说出这行文本的颜色 </div>
  </div>
</div>
```

运行以后，浏览器中这行文本将显示__________色。

(3) 某 div 元素的宽度为 270px，它有一个子元素宽度设置为 10%，那么这个子元素在页面中显示的宽度为__________px。

(4) 在一个样式表文件 bg.css 中设置了 body 元素的背景色，代码如下:

```
body {
    background-color:rgb(0%, 0%, 100%); /* 蓝色 */
}
```

然后在页面中对 body 进行了一次背景色的设置，并引入了 bg.css 样式表文件，代码如下:

```
<style type="text/css">
body {
    background-color:#0f0; /* 绿色 */
}
@import "bg.css";
</style>
```

那么，最后浏览器中页面背景色将为__________色。

2. 选择题

(1) 在下面关于 CSS 样式的说明，其中__________不是 CSS 的优势。

A. Web 页面样式与结构分离

B. 页面下载时间更快

C. 轻松创建及编辑

D. 使用 CSS 能够增加维护成本

(2) 使用 CSS 样式表对页面设置背景色为 rgb(255, 0, 0)，那么在浏览器中页面的背景色将

为__________色。

A. 红　　B. 蓝　　C. 绿　　D. 黄

(3) 长度单位 px 是相对于__________的，所以在计算机中通常以它作为长度衡量单位。

A. 屏幕大小　　B. 显示器分辨率

C. 电脑配置　　D. 打印尺寸

(4) 下面定义的 CSS 样式，__________不能正常设置页面背景图片。

A.

```
body {
    background-image:url("gb.jpg");
}
```

B.

```
body {
    background-image:url(gb.jpg);
}
```

C.

```
body {
    background-image:url('gb.jpg');
}
```

D.

```
body {
    background-image:gb.jpg;
}
```

(5) 页面中有以下代码:

```
<style type="text/css">
div{ color:blue; /* 蓝色 */ }
#d1 #d2 div{ color:yellow; /* 黄色 */ }
#d1 div{ color:red; /* 红色 */ }
div#d3{ color:green; /* 绿色 */ }
</style>

<div id="d1">
  <div id="d2">
    <div id="d3"> 请说出这行文本的颜色 </div>
  </div>
</div>
```

在浏览器中运行以后，在页面中这行文本的颜色为__________。

A. 蓝色　　B. 黄色　　C. 红色　　D. 绿色

3. 上机练习

上机练习：使用 Dreamweaver 创建一个简单的页面

通过本章的学习，我们已经对 CSS 有了一个比较全面的了解。而且我们在本章中介绍了几种网页制作工具。

本练习我们就使用 Dreamweaver 创建一个简单的页面，在 Dreamweaver 中使用外部样式表为页面设置一张背景图片，并更改页面中文本的字体颜色，如图 1-10 所示。

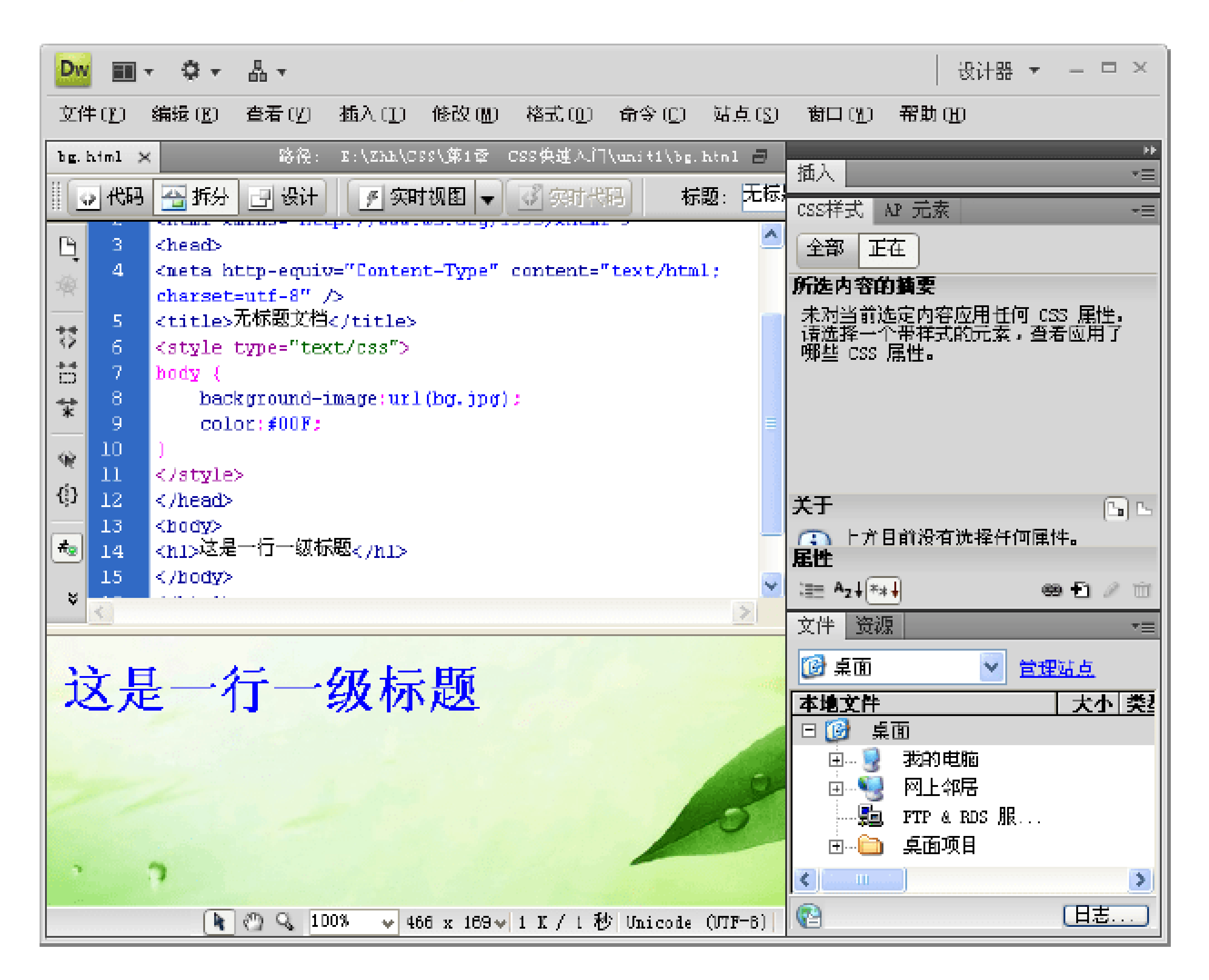


图 1-10　使用 Dreamweaver 编辑网页



第 2 章　选　择　符

内容摘要：

任何 HTML 元素都可以是一个 CSS 的选择符。选择符仅仅是指向特别样式的元素，例如选中页面中指定 id 的元素之后，就可以对这个元素的样式或者事件进行设置，这种选择符就叫做 id 选择符，这只是选择符中的一种。选择符还包括通用选择符、类型选择符、class 选择符、伪类及伪元素选择符、群组选择符和包含选择符等。

本章将讲解 CSS 选择符的使用。

学习目标：

- 了解并掌握通用选择符
- 了解并掌握类型选择符
- 了解并掌握 id 选择符
- 了解并掌握 class 选择符
- 了解并掌握伪类及伪元素选择符
- 了解并掌握群组选择符
- 了解并掌握包含选择符

2.1 设置指定文字的样式

在 CSS 样式中，如果要选择所有具有共同属性的标签元素，可以使用通用选择符。如果要选择页面中所有的标签元素，可以使用“*”选择符。

 **视频教学：**光盘/视频/2/2.1 设置指定文字的样式.avi 长度：4 分钟

2.1.1 基础知识——通用选择符

通用选择符用于从文档中获取所有元素。它的基本语法如下：

```
*{sResult}
```

使用通用选择符时，通常可结合上下文来搜索查找元素。假如通用选择符不是单一选择符中的唯一组成，“*”可以省略。

例如，下面实例代码：

```
<p>p 元素</p>
<div>div 元素</div>
```

设置 p 元素和 div 元素的字体颜色为红色，CSS 代码如下：

```
*{ color:red;}
```

2.1.2 实例描述

在 HTML 页面中，免不了有很多同样的标签，如果有些标签具有同一个 CSS 样式，那么就不能一个一个地给它们设置样式了，因为那样会很繁琐，使用通用选择符就能解决这个问题。

2.1.3 实例应用

【例 2-1】设置指定文字的样式

(1) 新建一个 HTML 页面，名称为 tongyong.html。

(2) 现在页面中有 5 个 p 元素，代码如下所示：

```
<div id="p_div">
    <p>一曰天伦乐  夫妻互敬互爱，兄弟亲热相好，妯娌和睦相处，儿女孝顺父母，长辈关心晚辈，经常让属于自己的小天地充满温馨。</p>
    <p>二曰运动乐  若要健，天天练，运动是健康的源泉；活动好比灵芝草，何必苦把仙方找；年轻跳蹦蹦，到老没病痛；身体锻炼好，八十不服老。</p>
    <p lang="af">三曰聊天乐</p>
    <p>四曰互助乐  鱼水生存，人人活着，我为人人，人人为我。一人有难大家帮，一家有事百家忙；我帮别人要忘却，别人帮我记心上。</p>
```

```
    <p lang="af">五曰宽容乐</p>
</div>
```

(3) 将所有 lang 属性值为 af 的 p 元素的字体颜色设置为红色，字体大小为 30px，CSS 代码如下：

```
*[lang=af] { color:red; font-size:30px;}
```

(4) 如果将所有的段落字体设置为蓝色，宽度为 400px，CSS 代码如下：

```
*{ color:#06F; width:400px;}
```

2.1.4　运行结果

打开该网页，效果如图 2-1 所示。



图 2-1　通用选择符

2.1.5　实例分析



源码解析：

该案例中，有 5 个 p 元素，首先将所有 lang 属性值为 af 的 p 元素的字体大小设置为 30px，字体颜色为红色，选择目标元素的代码如下：

```
*[lang=af]
```

当然，也可以根据标签的其他属性来进行筛选。

2.2　设置表格的样式

表格的样式，如表格的宽度、高度、表格字体及字体颜色等这些可设置的属性，都可以通过使用标签选择符来设置样式。这个小节主要讲解关于标签选择符的使用。

 **视频教学：**光盘/视频/2/2.2　设置表格的样式.avi　　　　长度：5 分钟

2.2.1　基础知识——标签选择符

HTML 页面中的标签元素，如 td 元素、li 元素等，可以作为标签选择符来使用。标签选择符的语法如下：

```
element{}
```

其中，element 表示可选的标签元素。使用标签选择符能够为相同的标签定义不同的样式。例如，下面实例代码：

```
<p>p 元素一</p>
<p>p 元素二</p>
<div>div 元素</div>
```

将 p 元素的字体颜色设置为红色，CSS 代码如下：

```
p{ color:red;}
```

2.2.2　实例描述

上小学的时候，我经常会在语文书上画一个非常显眼的表格——课程表，表格里面有红色字体和黑色字体，还挺好看。下面的这个例子，就以表格为例，讲述类型选择符的使用。

2.2.3　实例应用

【例 2-2】设置表格的样式

(1) 新建一个 HTML 页面，名称为 leixing.html。

(2) 在页面中制作一个小学生课程表，HTML 代码如下：

```
<table width="433" height="165" border="1">
  <tr>
    <td width="54">课程表</td>
    <td width="70">星期一</td>
    <td width="70">星期二</td>
    <td width="57">星期三</td>
    <td width="72">星期四</td>
```

```
    <td width="70">星期五</td>
  </tr>
  <tr>
    <td>上午</td>
    <td>语文/数学</td>
    <td>数学/语文</td>
    <td name="red">微机课</td>
    <td>语文/数学</td>
    <td>数学/语文</td>
  </tr>
  <tr>
    <td>下午</td>
    <td>数学/语文</td>
    <td>语文/数学</td>
    <td name="red">体育课</td>
    <td>数学/语文</td>
    <td>语文/数学</td>
  </tr>
</table>
```

(3) 设置表格的颜色可以用 table 元素，也可以直接用 td 元素，这里用的是 td 元素；设置单元格的字体颜色，可以使用 td 元素，也可以设置指定属性的 td 元素的样式，CSS 代码如下：

```
td{ background-color:#CC3;}
body{ background-image:url(http_imgload6.jpg); background-repeat:no-repeat;}
table{ margin-left:130px; margin-top:150px;}
td{ color:black;}
td[name="red"]{ color:red;}
```

2.2.4　运行结果

打开网页，效果如图 2-2 所示。



图 2-2　设置表格的样式

2.2.5　实例分析


源码解析：

使用标签选择符可以一次匹配所有标签的样式，但是却无法为每个标签定义独立的样式。在以上案例中，用 td 选择符设置单元格的字体颜色，会将所有的单元格设置为同一个样式。CSS 代码如下：

```
td{ background-color:#CC3;color:#06F;}
```

如果要设置指定单元格的样式，可以在标签选择符后面加上指定元素的属性及属性值，CSS 代码如下：

```
td[name="red"]{ color:red;}
```

2.3　实现图片的层叠效果

当从电脑桌面上打开多个窗口时，这些窗口会层叠到一起，但这不会影响查看效果。因为要查看的窗口始终处于激活状态，且它的层次是最高的。

本小节将讲解如何用 ID 选择符实现图片层叠的效果。


视频教学：光盘/视频/2/2.3　实现图片的层叠效果.avi　　　　长度：5 分钟

2.3.1　基础知识——id 选择符

在 HTML 页面中 id 参数指定了某个单一元素，id 选择符用于对这个单一元素定义单独的样式。id 选择符的语法如下：

```
#id{}
```

和类选择符不同，id 选择符通常用来选择一个特定的元素，而非一组元素。为了通过 id 选择符对目标元素的引用，为 id 添加了一个前缀“#”。

例如，下面实例代码：

```
<input id="txt" type="text" value="admin" />
```

使用 id 选择符将<input>标签的字体颜色设置为红色，CSS 代码如下：

```
#txt{ color:red;}
```

2.3.2　实例描述

在我的面试经历中，曾经有一个经理面试时只提了一个问题，就是如何实现层叠效果。我

的回答是用绝对定位，但这位经理说不知道什么是绝对定位，你就告诉我你怎么实现层叠效果，我的回答还是用绝对定位，用 position 属性。我后来又一想，用 position 属性是没错，但是实现层叠效果最关键的是 z-index 属性，单纯地设置 position 属性并不能实现层叠的效果……

好了，说了这么多，就是为下面的这个例子做铺垫，以加深读者的记忆。

2.3.3　实例应用

【例 2-3】实现图片的层叠效果

(1) 新建一个 HTML 页面，名称为 id.html。

(2) 页面中有两个<div>标签，id 分别为 top-div、bottom-div。在每个<div>标签中分别嵌套一个图片，HTML 代码如下：

```
<div id="top-div">
   <img src="http_imgload17.jpg" />
</div>
<div id="bottom-div">
   <img src="3.jpg" height="258"/>
</div>
```

(3) 现在，要实现让上面定义的两个<div>标签层叠起来。具体为：id 为 top-div 的<div>标签在下面，id 为 bottom-div 的<div>标签在上面。CSS 实现代码如下：

```
#top-div{ background-color:#FCF; position:absolute; z-index:1;}
#bottom-div{ background-color:#0CF; margin-top:200px; margin-left:80px;
position:absolute; z-index:2; filter:alpha(opacity=40);}
```

2.3.4　运行结果

打开网页，效果如图 2-3 所示。



图 2-3　实现层叠效果

2.3.5 实例分析


源码解析：

id是唯一的，用“#”来表示id选择符。在该案例中创建了两个div标签，每个div标签都有一个id。id值不同，因此也说明了id是唯一的。根据这两个div的id设置各自的CSS样式，即它们的层叠次序z-index的值等。

2.4 克隆div元素

所谓克隆div元素，就是将所有div元素的CSS样式设置为相同。本节以class选择符为主题讲解如何克隆div元素。


视频教学：光盘/视频/2/2.4 克隆div元素.avi　　长度：5分钟

2.4.1 基础知识——class选择符

class选择符的语法如下：

```
.ClassName{}
```

其中，这个圆点是英文字符“.”。一个类名可以应用到多个页面元素上，当然一个页面元素也可以应用多个类名。

参考以下实例：

```
div.note{font-size:14px;}
```

表示所有class属性值等于(包含)“note”的div对象字体尺寸为14px。

```
.dream{font-size:14px;}
```

表示所有class属性值等于(包含)“dream”的对象字体尺寸为14px。

2.4.2 实例描述

在网页中，经常会有多个页面模块使用同样的字体、字号、颜色、边框、背景等。而如果为每一个页面模块都重复定义一次CSS样式，那么将会非常的费时费力。

CSS中提供了一个为同一类元素分组设置样式的机制，那就是class选择符。下面通过一个实例来看一下class选择符的用法。

2.4.3　实例应用

【例 2-4】克隆 div 元素

(1) 新建一个 HTML 页面，名称为 class.html。

(2) 页面中有 4 个<div>标签，并且都引用了同一个类名，代码如下：

```
<div class="div" id="div1">
莫想老，常思少，思想不老化，衣着老来俏。不拘年岁行辈多与青年人交朋友，萌发童心有奇效；人老心不老，黄昏爱恋讲科学，求得“第二青春”之愉悦。
</div>
<div class="div" id="div2">
知识是社会的宝贵财富，是人们的精神食粮。多看书、常读报既增见识，长才干，又驱寂寞，益身心。
</div>
<div class="div" id="div3">
适度的体力劳动，可使五脏功能得到锻炼，减缓衰老。实践证明劳者寿。
</div>
<div class="div" id="div4">
鱼水生存，人人活着，我为人人，人人为我。一人有难大家帮，一家有事百家忙；我帮别人要忘却，别人帮我记心上。
</div>
```

(3) 将这 4 个<div>标签的样式设置为一样的，而它们都将调用类名为 div 的 CSS 样式，CSS 代码如下：

```
body{ background-image:url(http_imgload6.jpg); background-repeat:no-repeat;}
.div{ width:150px; height:150px; background-color:#FF9;float:left; margin-left:2px;
border:#C3C 2px solid; margin-top:180px;}
```

2.4.4　运行结果

在浏览器中打开该网页，效果如图 2-4 所示。

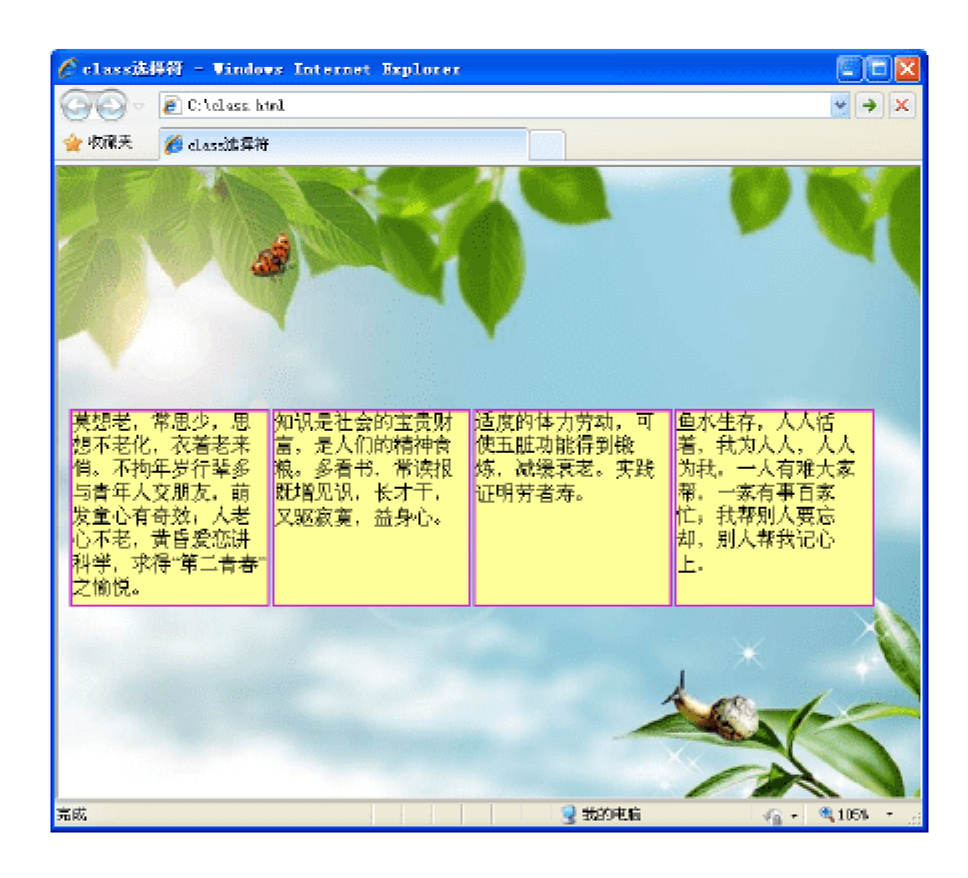


图 2-4　克隆 div 元素

2.4.5　实例分析



源码解析：

类选择符可以应用于多个页面元素，以英文圆点“.”作为类选择符，去设置页面元素的样式。如在该案例中，有 4 个<div>标签，将类名为 div 的样式应用于这 4 个<div>标签上。代码如下：

```
.div{ width:150px; height:150px; background-color:#FF9;float:left; margin-left:2px;
border:#C3C 2px solid; margin-top:180px;}
```

2.5　实现动态变色的效果

所谓动态变色，就是当鼠标移上去时改变目标元素的背景颜色，鼠标移下来就会变回目标元素原来的背景色。在这个小节中，我们就一起来实现动态变色的效果，顺便学习什么是伪类及伪元素选择符。



视频教学：光盘/视频/2/2.5　实现动态变色的效果.avi　　　　长度：5 分钟

2.5.1　基础知识——伪类及伪元素选择符

伪类选择符可以看作某种特殊的类选择符，是能被支持 CSS 的浏览器自动识别的特殊选择符。伪类选择符的最大作用就是可以对链接的不同状态定义不同的样式效果。伪类选择符定义的样式最常应用在定位锚标记(<a>)上，即锚的伪类选择符。它表示动态链接 4 种不同的状态：未访问的链接(link)、已访问的链接(visited)、激活链接(active)和鼠标停留在链接上(hover)。

伪类选择符的语法如下(以:link 伪类为例)：

```
Selector:link{}
```

伪元素选择符的语法如下(以:before 伪元素为例)：

```
Selector:before{}
```

其中，参数 Selector 表示目标元素，后面的表示伪类或伪元素。

2.5.2　实例描述

伪类和伪元素选择符可以很轻松地为特定元素添加一组样式，它实现了以前必需使用 JavaScript 配合才能够显示出来的特殊页面效果。下面我们就来使用伪类和伪元素选择符实现一个鼠标经过改变样式的页面特效。

2.5.3　实例应用

【例 2-5】实现动态变色的效果

(1) 新建一个 HTML 页面，名称为 weilei.html。

(2) 页面中有一篇散文，代码如下：

```
<div id="title_div">
  <h3>花开的声音</h3>
</div>
<div id="main_div">
  <p>抬头看看天，我无奈地叹口气。为了躲避父母的轮番轰炸，我慌称要去参加周末辅导班。现在的我最怕呆在家里，最怕看到爸爸若有所思的眼神，还有妈妈手里端着的人参鸡汤。骑着自行车，我慢无目的地逛着，我也不知道自己想干什么。去找她吧！她是谁？我也不知道，她在网上，叫花开的声音，是我现在唯一的朋友。抛开所有的唠叨和叮嘱，我又来到了“深蓝色”网吧。
    </p>
  <p>
        你愿意振作起来，去实现自己的梦想吗？我的明天是蓝色的，因为是我自己选的，所以不管它是什么样子，对我来说都是最美的。你为你的明天选好颜色了吗？大家都相信你，是因为他们爱你，因为你会努力。我也相信你，因为你是“超人”。
  </p>
</div>
<p>
  把信封好，我感到快乐。翻开厚厚的习题册，我又开始新一轮英语自测。最后的成绩”119 分。我没有灰心，
下一次会更好。这时妈妈推开门进来，“喝牛奶吗。”我点点头，看着她眼角的皱纹说：“谢谢妈妈。”我用眼神
告诉她----“我会努力的！”
</p>
```

(3) 鼠标移到 div 元素上面，将改变 div 的背景色和鼠标的样式，用伪类选择符 div:hover 去实现它，CSS 代码如下：

```
body{ background-image:url(http imgload6.jpg); background-repeat:no-repeat;}
div:hover{ background-color:#F96; cursor:pointer;}
#title_div { margin-left:200px; margin-top:100px; width:100px;}
#main_div{  width:600px; margin-left:50px; margin-top:20px;}
p{ text-indent:2em}
```

2.5.4　运行结果

在浏览器中打开该网页，效果如图 2-5 所示。



图 2-5　页面动态变色前

鼠标移到文字上面时，改变文字的背景颜色，效果如图 2-6 所示。



图 2-6　鼠标移上去

2.5.5　实例分析



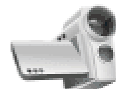
源码解析：

该案例通过讲解伪类及伪元素选择符来实现页面的一些动态效果。在这里使用了:hover 这个伪类，实现了鼠标移上去变色，移下来失色的动态变色效果。关键代码如下：

```
div:hover{ background-color:#F96; cursor:pointer;}
```

2.6　设置统一样式

指定将页面中不同的元素设置为统一样式，例如设置同样的背景色，设置同样的字体及字体大小等。在 Web 开发中，页面中的元素很多，但是有些元素需要使用同一个样式，那么我们就可以把这些元素取出来，形成群组选择符并设置统一样式。

**视频教学：** 光盘/视频/2/2.6　设置统一样式.avi　　　　长度：5 分钟

2.6.1　基础知识——群组选择符

将同样的定义应用于多个选择符，可以将选择符以逗号分隔的方式并为组。基本语法如下：

```
E1,E2,E3{}
```

例如，页面中有 table 元素、div 元素和 a 元素，代码如下：

```
<table>
   <tr>
     <td></td>
     <td class="td1"></td>
     <td></td>
   </tr>
</table>
<div><a href="#">海底世界</a></div>
```

将 class 属性值为“td1”的 td 元素和<div>标签中包含的 a 元素的字体大小设置为 14px，CSS 代码如下：

```
.td1,div a{ font-size:14px;}
```

2.6.2　实例描述

在我们的日常生活中，最不可缺少的就是日常用品了，例如吃的、用的等。

在一个风和日丽的星期天，我和朋友去逛超市，里面最吸引我的就是吃的东西了。在超市的货架上，有很多物品有着同样的包装，不同物品也有着同样的价格。

下面的这个小例子，就讲解如何将多个不同的页面元素设置为统一样式。

2.6.3　实例应用

【例 2-6】 设置统一样式

(1) 新建一个 HTML 页面，名称为 fenzu.html。

(2) 页面中有一个表格，用于存放手机的相关信息，例如下面的代码：

```
<TR>
  <TD height=30 bgcolor="#F3F7FA">手机品牌 ：<font color="#FF0000">CECT</font> </TD>
</TR>
<TR>
  <TD height=30 bgcolor="#F3F7FA">手机型号 ：<font
color="#FF0000">V958</font></TD>
</TR>
```

(3) 如上代码所示，设置<td>标签、<a>标签和<input>标签的字体颜色和边框样式等，CSS 实现代码如下：

```
td,td a,input{ color:#06F; font-size:14px; border:#9C3 solid 1px;}
```

2.6.4 运行结果

在浏览器中打开该网页，效果如图 2-7 所示。

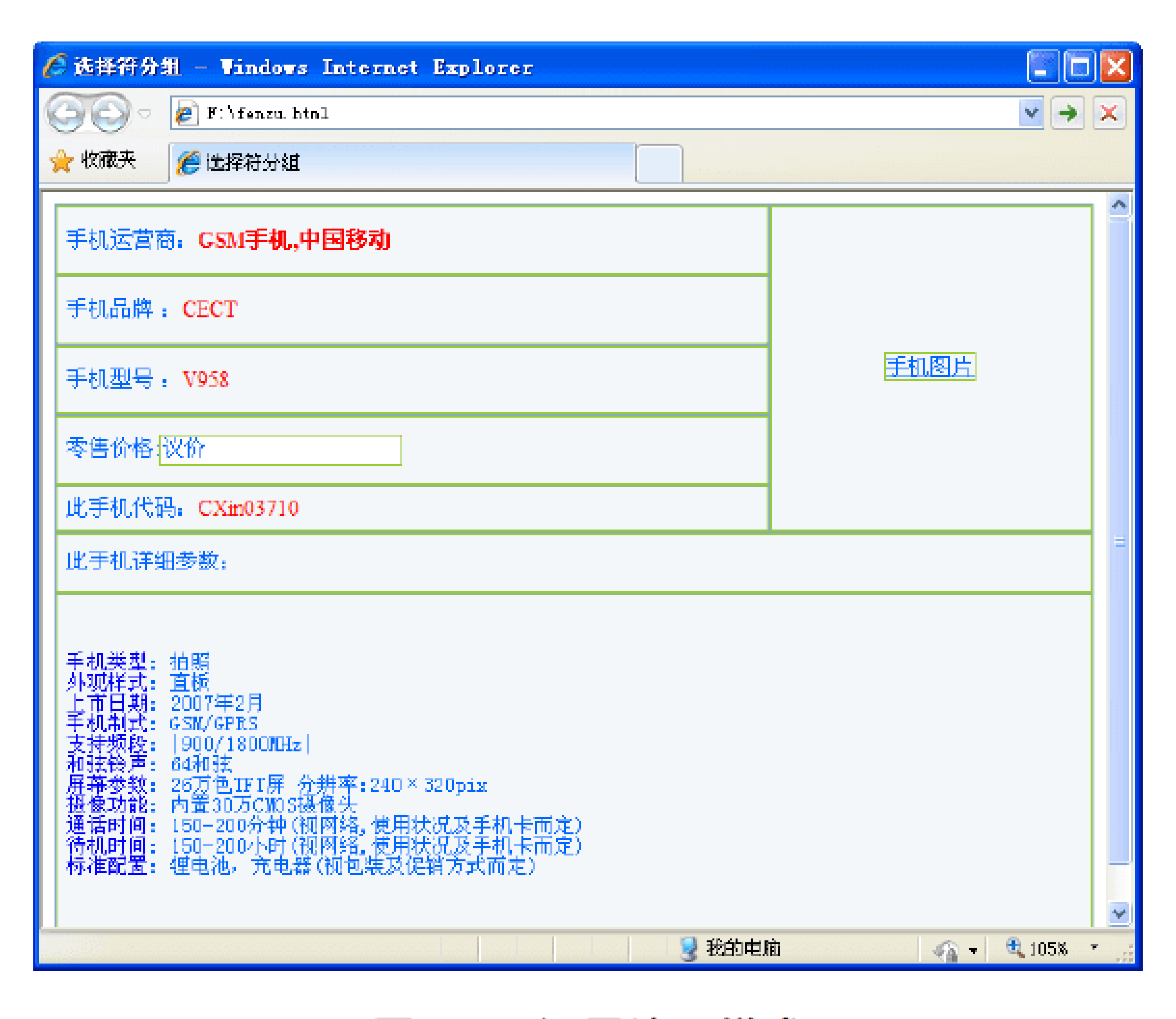


图 2-7　设置统一样式

2.6.5 实例分析



源码解析：

在这个案例中，主要讲解了什么是群组选择符，如何使用群组选择符对某些指定的标签设置统一样式。例如在该案例中，设置<td>标签和<input>标签的边框样式、字体颜色和大小，关键代码如下：

```
td,td a,input{ color:#06F; font-size:14px; border:#9C3 solid 1px;}
```

2.7　设置 div 中包含的标签样式

在 Web 网页的开发过程中，要想整个页面看起来整齐又好看，常常会用 div 元素将整个页面的元素进行布局，例如在一个 div 标签中，可以嵌套 div 元素、p 元素、input 元素等一些表单元素，也可以使用包含选择符来设置这些嵌套元素的样式。

**视频教学：**光盘/视频/2/2.7　设置 div 中包含的标签样式.avi　　长度：4 分钟

2.7.1　基础知识——包含选择符

包含选择符用于获取当前元素的指定子元素。其语法如下：

```
E1 E2{}
```

参考以下实例：

```
table td { font-size:14px; }
```

表示将 table 里面 td 元素的字体大小设置为 14px。

```
div.sub a { font-size:14px; }
```

表示将 class 属性值为 sub 的 div 元素下的 a 元素字体大小设置为 14px。

与包含选择符作用一样的就是子对象选择符，子对象选择符的语法如下：

```
E1>E2{}
```

表示 E2 必须是 E1 的直接子元素，而包含选择符的目标元素则只需要包含当前元素。例如下面的这个例子。

```
<div>
   <ul>
      <li> </li>
   </ul>
<div>
```

要定义 li 的样式，用包含选择符实现，代码如下：

```
div li{...}
```

用子对象选择符实现，代码如下：

```
ul>li{...}
```

而不能使用如下代码：

```
div>li{...}
```

2.7.2　实例描述

包含选择符可以选定某个容器元素内部的特定子元素，然后为其设置 CSS 样式。例如这里我们要将一个 div 中的超链接和指定页面元素设置相应的 CSS 样式，就可以使用本节所讲的包含选择符。

2.7.3　实例应用

【例 2-7】设置 div 中包含的标签样式

(1) 新建一个 HTML 页面，名称为 baohan.html。

(2) 页面中有一个 div 元素，div 里面嵌套了两个 input 元素和两个 a 元素，设计一个简单的用户登录界面，其实现代码如下：

```
<div>
   用户名：<input type="text" value="admin" />
   <br />
   <br />
   密码：<input type="password" value="admin" />   <a id="find" href="#">忘了密
码？请单击这里</a>
   <br />
   <br />
   <a id="sub" href="#">提交</a>
</div>
```

(3) 为<div>标签里面的不同元素设置不同的样式，例如设置不同的字体、字体大小、字体颜色或者外边距，实现代码如下：

```
body{ background-image:url(http_imgload6.jpg); background-repeat:no-repeat;}
 div{ margin-left:200px; margin-top:200px;}
 div #sub{color:#693; font-size:20px;}
 div input{ color:#03C;}
 div #find{ color:#96C; font-style:italic;font-size:20px;}
```

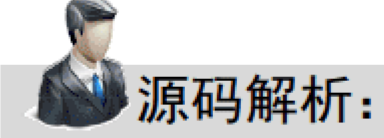
2.7.4　运行结果

在浏览器中打开该网页，效果如图 2-8 所示。



图 2-8　设置 div 中包含的标签样式

2.7.5　实例分析



源码解析：

在这个简单的登录界面中，为 div 元素中的不同元素设置不同的 CSS 样式，例如将 id 为 sub 的 a 元素的字体颜色设置为#693，字体大小设置为 20px，实例代码如下：

```
div a{color:#693; font-size:20px;}
```

2.8　常见问题解答

2.8.1　CSS 中的包含选择符与子对象选择符的区别



请问 CSS 中的包含选择符与子对象选择符有什么区别？

网络课堂：http://bbs.itzcn.com/thread-12332-1-1.html

请问 CSS 中的包含选择符与子对象选择符有什么区别？

【解决办法】举个例子：

```
<div>
   <ul>
      <li> </li>
   </ul>
<div>
```

要定义 li 的样式，用包含选择符，写成：

```
div li{...}
```

没有问题。

若用子选择符，只能写成：

```
ul> li{...}
```

不能写成：

```
div> li{...}
```

因为 div 和 li 非“父子关系”，而是“祖孙关系”。子对象提供了更精细的控制，由此可见包含选择符的深度和广度超过子对象选择符；而子对象选择符的针对性和唯一性比包含选择符强。

2.8.2 CSS 选择符问题



CSS 选择符问题。

网络课堂：http://bbs.itzcn.com/thread-12333-1-1.html

请问 p#submit 是什么意思？注意我说的 p 与#submit 之间没空格？

【解决办法】p#submit 的意思是：

```
<p id="submit">我是例子一</p>
```

p #submit 的意思是：

```
<p>
 <span id="submit">我是例子二</span>
    </p>
```

这两种用法的区别是：第一个是指 id 属性为 submit 的 p 标签，第二个是指 p 标签下一级的 id 属性为 submit 的元素。

2.8.3 怎样用 CSS 选取表格的第一列 CSS 选择符



怎样用 CSS 选取表格的第一列 CSS 选择符？

网络课堂：http://bbs.itzcn.com/thread-12334-1-1.html

我想让表格的第一列右对齐：

```
...
<style type="text/css">
.col_1 {text-align: right;}
</style>
...
<table>
<tr><td class="col_1">第一列</td><td>第二列</td></tr>
<tr><td class="col_1">第一列</td><td>第二列</td></tr>
<tr><td class="col_1">第一列</td><td>第二列</td></tr>
```

```
<tr><td class="col_1">第一列</td><td>第二列</td></tr>
<tr><td class="col_1">第一列</td><td>第二列</td></tr>
</table>
...
```

但是如果表格行数很多，若每行都写 class="col_1"，代码会很臃肿，有其他合适的方法吗？

【解决办法】建议用 th。

```
<style type="text/css">
.m_tb th{text-align:right;font-weight:normal;}
</style>
<table class="m_tb">
<tr><th>第一列</th><td>第二列</td></tr>
<tr><th>第一列</th><td>第二列</td></tr>
<tr><th>第一列</th><td>第二列</td></tr>
</table>
```

2.9　习　　题

1. 填空题

(1) 选择页面中的所有标签元素，除了用 body 之外，还可以使用__________。

(2) 使用类型选择符来设置 table 表格的字体颜色，代码是__________。

(3) id 选择符是以文档目录树(DOM)中作为对象的唯一标识符的________作为选择符。

(4) 使用__________能够为相同的标记定义不同的样式。

(5) 使用__________可以设置多个不同元素的样式。

2. 选择题

(1) 以下__________选项属于 CSS 选择符。

A. 超文本标记选择符　　B. id 选择符
C. 类选择符　　D. 包含选择符

(2) 在 Dreamweaver 中，关于用 z-index 改变层的次序，说法正确的是__________。

A. z-index 的值越大，这个层的位置就越靠上
B. z-index 的值越大，这个层的位置就越靠下
C. z-index 的值越大，这个层的位置就越靠中央
D. 以上说法都错

(3) 下列__________选项是群组选择符的语法。

A. E1 E2{}　　B. E1,E2{}　　C. E1>E2{}　　D. *{}

(4) 以下__________选项是伪类选择符。

A. .class　　B. #id　　C. #id a　　D. #id:hover

(5) 下列__________选项是包含选择符。

A. E1 E2{}   B. E1,E2{}   C. E1>E2{}   D. *{}

3. 上机练习

上机练习：设置一个表格的样式

创建一个表格，并设置该表格的样式，效果如图 2-9 所示。

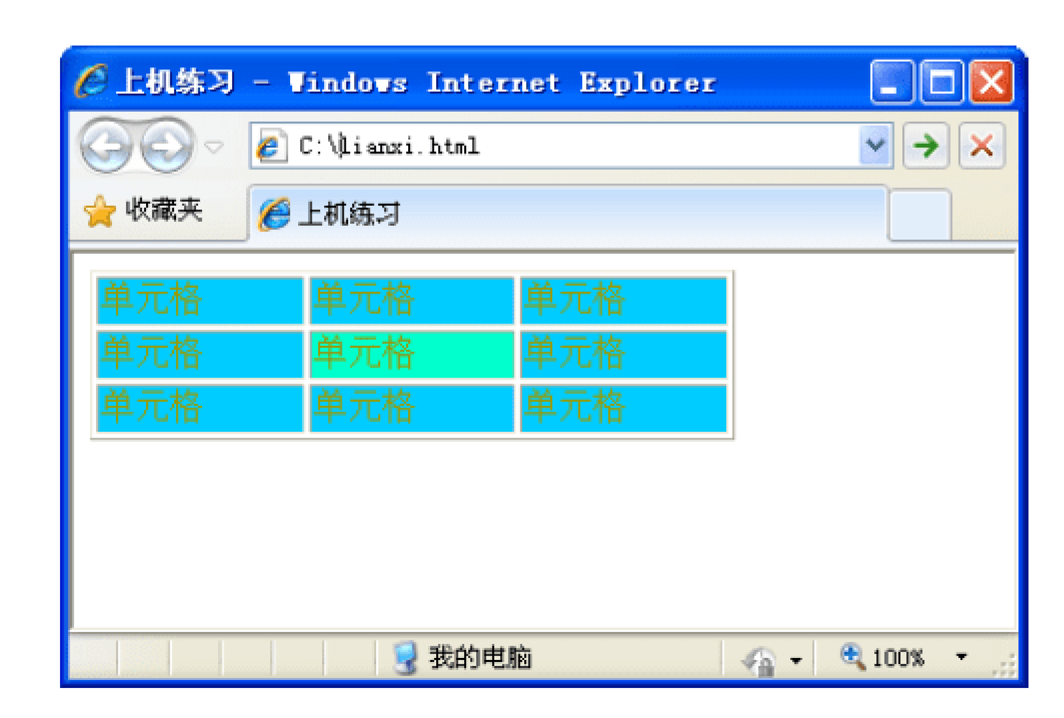
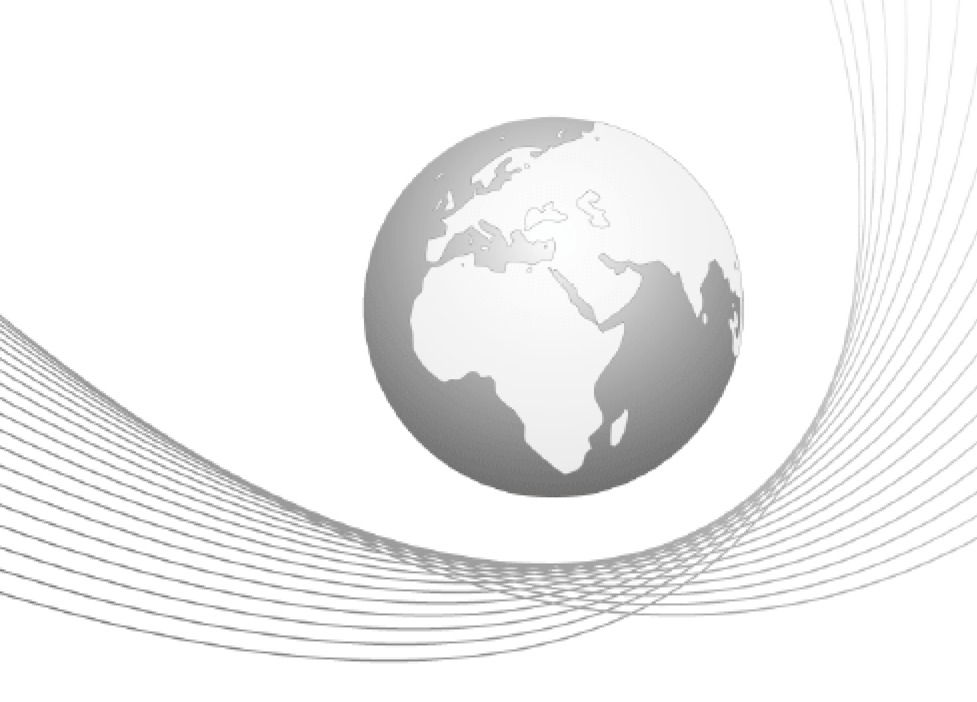


图 2-9　设置表格的样式



第 3 章　伪类和伪元素

内容摘要：

伪类可以看做是一种特殊的类选择符，是能被支持 CSS 的浏览器自动识别的特殊选择符，也就是不用再使用 class 或 id 调用就可以直接使用的，它的最大用处就是可以对链接定义不同状态下的样式效果。

CSS 中伪元素的作用也是如此，它可以将特殊的效果添加到某些选择符。

在本章中我们将会详细讲解每一个 CSS 伪类和伪元素的用法，并结合实际案例帮助读者快速掌握。

学习目标：

- 掌握链接伪类的使用方法
- 熟悉:focus 伪类的使用
- 了解:first-child 伪类的使用
- 了解适用于@page 的伪类
- 熟悉:lang 伪类的使用
- 了解:first-letter 伪元素和:first-line 伪元素
- 了解:before 伪元素和:after 伪元素

3.1　制作多彩的超链接

在网页中，默认的文字链接样式都是带下划线的效果，这种一成不变的外观太单一，也不符合页面的整体效果，CSS 样式可以改变这种单一形式。

通过使用 CSS 伪类，可以设置文字链接在不同状态下的不同样式显示，例如文字颜色、样式、有无下划线等。

**视频教学：**光盘/视频/3/3.1　制作多彩的超链接.avi　　长度：6 分钟

3.1.1　基础知识——链接伪类

链接伪类是众多 CSS 伪类中使用频率最多的，在学习它的应用之前首先来了解一下伪类的基本语法。

伪类语法格式如下：

```
selector : pseudo-class {property: value}
```

CSS 类也可与伪类搭配使用，这种方式的语法格式如下：

```
selector.class : pseudo-class {property: value}
```

在支持 CSS 的浏览器中，链接的不同状态都可以使用不同的方式显示。这些状态包括选择状态、已被访问状态、未被访问状态和鼠标悬停状态。

例如，下面的实例代码：

```
a:link {color: #FF0000}          /* 未访问的链接 */
a:visited {color: #00FF00}       /* 已访问的链接 */
a:hover {color: #FF00FF}         /* 鼠标移动到链接上 */
a:active {color: #0000FF}        /* 选定的链接 */
```



在 CSS 定义中，a:hover 必须置于 a:link 和 a:visited 之后，才是有效的。同时，a:active 必须置于 a:hover 之后，才是有效的。

3.1.2　实例描述

在浏览网页时，经常看到这种效果：网页中的文字经鼠标触动，显示的样式各不相同，而且显得很符合整体页面的风格，效果非常酷。

其实，通过使用 CSS 伪类，很容易实现文字链接在不同状态下的不同样式显示。例如文字大小、有无边框或者下划线等。下面学习如何使用 CSS 制作多彩文字链接。

3.1.3　实例应用

【例 3-1】制作多彩的超链接

(1) 新建一个 HTML 页面，并添加一张背景图片。

(2) 在页面中添加如下的一行代码来创建一个 h1 元素，并包含一个超链接。

```
<h1><a href="#">小学教育</a></h1>
```

(3) 接下来为这个超链接定义样式，首先是未被访问过时的链接状态。这些 CSS 样式代码如下所示：

```
h1 a:link {
    color: #FF0000
}
```

(4) 定义被访问过时的链接样式。

```
h1 a:visited {
    color: #00FF00
}
```

(5) 定义鼠标移到链接上时样式。

```
h1 a:hover {
    color: #FF00FF
}
```

(6) 接下来定义鼠标在链接上按下时的样式。

```
h1 a:active {
    color: #0000FF
}
```

(7) 接上方的 h1 元素，这里再添加一个 ul 列表。

```
<ul class="main">
  <li><a href="#">数学 (2398)</a></li>
  <li><a href="#">语文 (2043)</a></li>
  <li><a href="#">英语 (1817)</a></li>
  <li><a href="#">奥数 (2491)</a></li>
  <li><a href="#">作文 (36) </a></li>
  <li><a href="#">科学 (1048) </a></li>
</ul>
```

(8) 为了更好地掌握链接伪类的用法。下面为 ul 列表中的链接定义样式，首先是链接的通用样式，这包括宽度、对齐方式、字体颜色、是否有下划线以及显示位置等，最终如下所示：

```
.main li a {
    color: black;
    text-decoration: none;
    padding-left: 20px;
```

```
    display: block;
    display: inline-block;
    width: 100px;
    padding-top: 3px;
    padding-bottom: 3px;
    text-align: left;
}
```

(9) 然后遵循为 h1 中链接定义样式的先后顺序，依次编写:link、:visited、:hover 和:active 4 个 CSS 样式，这部分代码如下所示：

```
.main li a:link {
    font-size: 14px;
    color: #A34B33;
    text-decoration: none;
    font-weight: bold;
}
.main li a:visited {
    font-size: 14px;
    color: #FF66CC;
    text-decoration: overline;
}
.main li a:hover {
    font-size: 14px;
    color: #000000;
    text-decoration: overline;
}
.main li a:active {
    font-size: 14px;
    color: #6633CC;
    text-decoration: line-through;
}
```

(10) 保存对 HTML 页面的修改，完成样式的定义。

3.1.4 运行结果

在浏览器中直接打开实例 HTML 文件，会先看到 h1 中链接的默认效果，如图 3-1 所示，它是由“h1 a:link”来定义的。



图 3-1 默认 h1 的链接效果

把鼠标移到其中的文字上，查看由“h1　a:hover”样式定义的效果，如图 3-2 所示。现在，按下鼠标查看“h1　a:active”样式定义的效果，如图 3-3 所示。



图 3-2　移到 h1 上时的链接效果



图 3-3　按下 h1 时的链接效果

下面，看看对 ul 列表中链接定义的 CSS 样式是否有效果。如图 3-4 所示为“.main li a:hover”样式移到上面时的效果，图 3-5 所示为“.main li a:active”中定义的被按下时的效果。

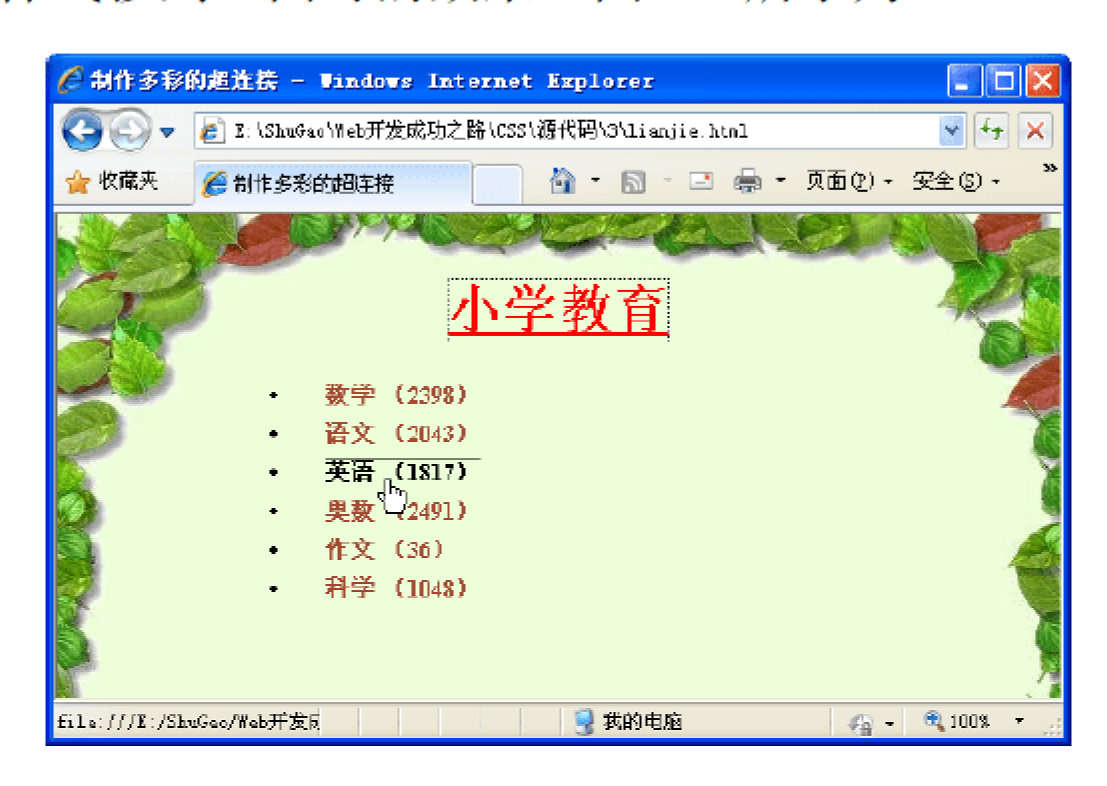


图 3-4　移到列表中链接上时的效果



图 3-5　按下列表中链接时的效果

3.1.5　实例分析



源码解析：

在本实例中，不仅为 h1 元素定义了各种链接状态时的显示样式。还可以通过选择符的组合，为 ul 列表中的每个链接定义样式。

在这样的链接伪类中可以使用任意有效的 CSS 样式修饰链接。唯一需要注意的是出现的顺序，必须遵循“a:link→a:visited→a:hover→a:active”顺序，否则就会显示样式无效，或者不能达到预期的效果。

3.2　高亮显示输入元素的登录表单

注册用户时，用鼠标单击输入用户名的“文本框”时，该“文本框”背景样式的显示与其他文本内容的样式不同；当浏览新闻时，用鼠标单击新闻标题，新闻标题与其他的内容有很大的差别，而鼠标离开，新闻标题则恢复正常。这样的效果就称为“高亮显示”。

相比之下，这些效果有没有吸引我们的注意呢？

 **视频教学：**光盘/视频/3/3.2　高亮显示输入元素的登录表单.avi　　　长度：5 分钟

3.2.1　基础知识——:focus 伪类

:focus 伪类用来设置对象在成为输入焦点，即触发该对象 onfocus 事件发生时的样式。该伪类的语法如下：

```
Selector : focus { sRules }
```

例如，下面的样式代码定义了链接成为焦点以及带图片链接成为焦点时的样式。

```
a:focus { font-size: 14pt; text-decoration: underline; color: blue; }
a:focus img { border: thin solid green }
```

3.2.2　实例描述

正如上面提到的，当输入用户名的文本框获得焦点时，文本框的样式发生改变，失去焦点时文本框的样式恢复到原来的样式。这个效果用 JavaScript 代码可以很容易地实现，但还有没有其他方法可以实现呢？答案是肯定的。

那就是使用本节介绍的:focus 伪类来实现，下面介绍它在登录表单中的应用吧。

3.2.3　实例应用

【例 3-2】高亮显示输入元素的登录表单

(1) 新建一个 HTML 页面，命名为 login.html。

(2) 在页面中添加一个 4 行 2 列的表格。

(3) 为表格填充内容，使其成为一个包含用户名文本框、密码文本框和“登录”按钮的表单。代码如下所示：

```
<table>
  <tr>
    <td colspan="2"  align="center">管理员登录</td>
  </tr>
```

```
  <tr>
    <td>用户名：</td>
    <td><input name="username" type="text" size="20" /></td>
  </tr>
  <tr>
    <td>密    码：</td>
    <td><input name="userpass" type="password" size="22" /></td>
  </tr>
  <tr>
    <td colspan="2" align="center"><input type="submit" value="登录" /></td>
  </tr>
</table>
```

(4) 在页面中用 CSS 的:focus 伪类来定义上述表单中 input 元素被触发时的样式。这里仅仅是修改了背景颜色，使其突出显示，代码如下所示：

```
input:focus {
    background-color:#6F3;
}
```

(5) 保存对 HTML 页面的修改，完成样式的定义。

3.2.4　运行结果

在浏览器中直接打开 login.html 文件，会看到默认的效果如图 3-6 所示。



图 3-6　默认效果

此时由于没有任何一个输入元素被激活，所以看不出效果。用鼠标单击“用户名”的文本框时，效果如图 3-7 所示，可以看到该文本框显示了在“input:focus”中定义的背景颜色。

如果单击“登录”按钮，也可以看到效果，如图 3-8 所示。

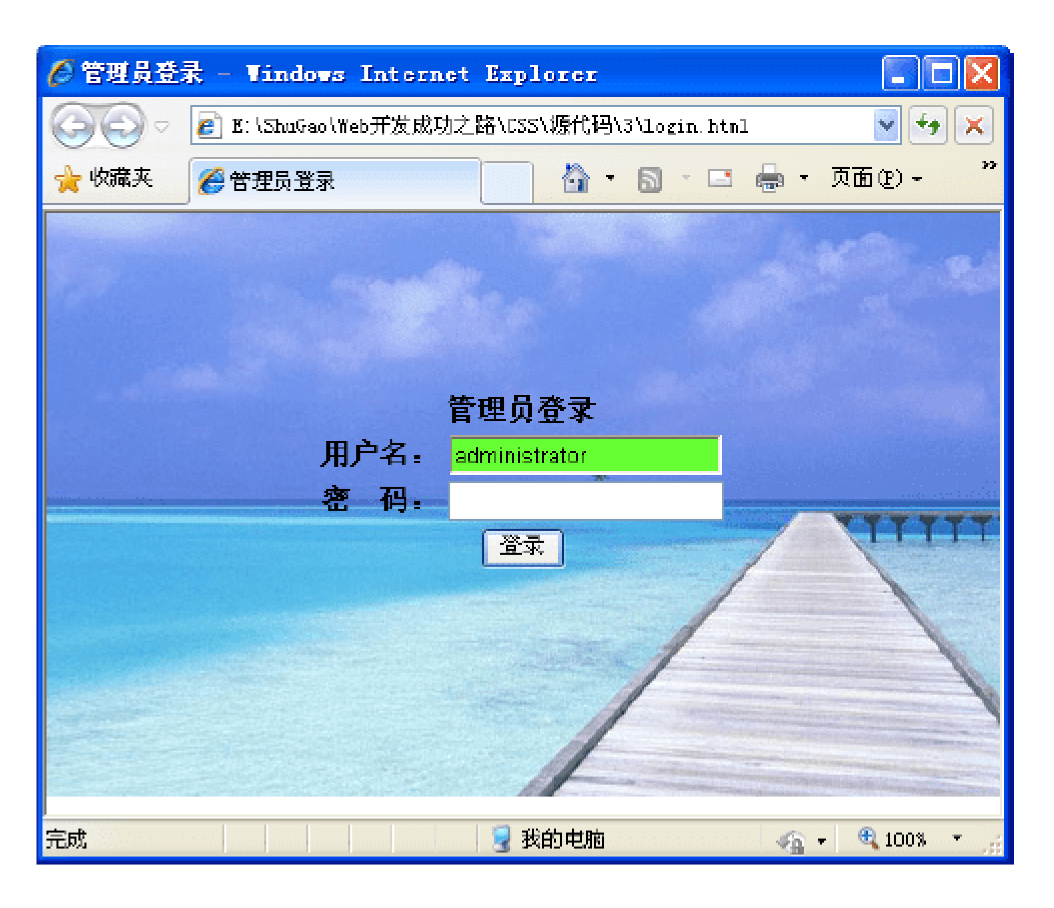



图 3-7　输入用户名时效果



图 3-8　单击按钮时效果

3.2.5　实例分析


源码解析：

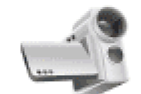
通过本实例的制作，可以看到使用:focus 伪类可以很简单地为鼠标当前激活的元素定义显示样式。

但是在使用时要注意，该伪类不支持 Internet Explorer 8 之前的版本，且在 Internet Explorer 8 中使用时必须已经定义了!DOCTYPE 元素，否则无效。其他浏览器都支持，无此问题。

3.3　为首个子对象定义样式

如果想要展示一组美观大方、样式统一的数据。你可能会说：先定义好一个样式，然后用 class 调用，那么有没有更简单，更方便的方法呢？肯定有，那就是要为首个子对象定义样式了。

首个子对象就是指某个元素下的第一个元素，例如，在代码“<div><p></p>
</div>”中，“<p></p>”为<div>下的第一个元素。


视频教学：光盘/视频/3/3.3　为首个子对象定义样式.avi　　　　长度：5 分钟

3.3.1　基础知识——:first-child 伪类

:first-child 伪类用来设置指定页面元素中第一个子对象的样式。该伪类的语法格式如下：

```
Selector : first-child { sRules }
```

:first-child 是一个很特殊的伪类，很容易遭到误解，所以有必要举例来说明。考虑以下元素：

```
<div>
  <p>栏目列表</p>
  <ul>
    <li>CSS 基础</li>
    <li>CSS 框模型<strong>最新</strong></li>
    <li>CSS 定位</li>
  </ul>
  <p>你也可以单击<em>添加</em>按钮，向网站中添加新内容。.</p>
</div>
```

在上面的例子中，作为第一个元素的元素包括第一个 p、第一个 li、strong 和 em 元素。

给定以下样式规则：

```
p:first-child {font-weight: bold;}
li:first-child {color:red;}
```

第一个规则将作为某元素第一个子元素的所有 p 元素设置为粗体。第二个规则将作为某个元素(在 HTML 中，这肯定是 ol 或 ul 元素)第一个子元素的所有 li 元素变成红色。



最常见的错误是认为 p:first-child 之类的选择器会选择 p 元素的第一个子元素。

3.3.2 实例描述

在使用传统的 CSS 时，都是用类样式或者 id 选择器来定义一个样式。

在掌握了:first-clid 伪类之后，可以利用它来设置样式，从而避免在很多类似的代码中频繁地调用“class”或者“id”，使代码更简洁。

3.3.3 实例应用

【例 3-3】为首个子对象定义样式

(1) 新建一个 HTML 页面，命名为 first-child.html。

(2) 在页面中添加一些布局代码，并注意它们的层次结构。例如，在本实例中使用如下代码制作了一个简单的课程列表。

```
<div>
  <p>窗内网 CSS 课程视频列表</p>
  <ul>
    <li>css 基础</li>
    <li>css 模型</li>
    <li>css 定位</li>
  </ul>
  <ul>
    <li>css 选择符</li>
```

```
    <li>css 属性</li>
    <li>css 伪类</li>
  </ul>
  <p>要了解更多有关 css 的资料，可以到<b>窗内网</b>。</p>
  <div>
    <p>版权所有&copy;itZcn.com</p>
  </div>
</div>
```

(3) 在这个课程列表中共出现了 3 个 p 元素。利用:first-child 伪类定义 p 元素第一次出现时的样式，这里为加粗、红色，代码如下所示：

```
p:first-child {
    font-weight: bold;
    color:red;
}
```

(4) 利用:first-child 伪类定义列表中 li 元素第一次出现时的样式，代码如下所示：

```
li:first-child {
    text-transform:uppercase;
    color:#00F
}
```

(5) 保存对 HTML 页面的修改，完成样式的定义。

3.3.4 运行结果

在浏览器中直接打开 first-child.html 文件查看运行效果，如图 3-9 所示。



图 3-9　运行效果

3.3.5　实例分析


源码解析：

通过本实例，相信读者对:first-child 伪类的使用有了进一步的认识。

但需要注意的是，:first-child 伪类仅会作用到同一层次的第一个对象上。例如，在本实例中，虽然 li 元素出现了 6 次，但由于有两个 ul 列表。所以，在每个 ul 列表中第一次出现 li 元素时将应用“li:first-child”样式。

对于 p 元素的应用也是同理，有两个 p 元素在同一个 div 内，因此只有第一次出现时将应用“p:first-child”样式。

3.4　设置 HTML 页面的打印尺寸

为了精确控制 HTML 页面的外观，CSS 不仅提供了在浏览器上显示的伪类，还可以通过:first 伪类、:right 伪类和:left 伪类来设置打印的尺寸。

下面，我们就一起来了解一下吧。


视频教学：光盘/视频/3/3.4　设置 HTML 页面的打印尺寸.avi　　　　长度：5 分钟

3.4.1　基础知识——:first 伪类、:right 伪类和:left 伪类

在 CSS 中提供了 3 个可适用于@page 规则的伪类，分别是:first、:right 和:left。

- :first

  :first 伪类用于设置页面容器第一页使用的样式，语法格式如下：

```
Selector : first { sRules }
```

  例如：

```
@page :first { margin: 4cm }
```

- :right

  :right 伪类用来设置页面容器位于装订线右边的所有页面使用的样式，语法格式如下：

```
Selector : right { sRules }
```

  例如：

```
@page :right { margin: 4cm }
```

- :left

  :left 伪类用来设置页面容器位于装订线左边的所有页面使用的样式，语法格式如下：

```
Selector : left { sRules }
```

例如：

```
@page :left { margin: 4cm }
```

3.4.2 实例描述

在网上冲浪时，经常会使用搜索引擎来查找自己比较感兴趣的内容。在找到之后，就会将它加入收藏夹，方便下次查看。

如果这些内容是资料性的，可能希望将它打印出来。此时，便会发现原本在 HTML 页面中正常显示的内容，打印出来之后发生了变化。这是由于打印的尺寸与显示尺寸不匹配所导致的。

解决的办法是利用 CSS 中为@page 规则提供的伪类来设置 HTML 页面的打印尺寸。

3.4.3 实例应用

【例 3-4】设置 HTML 页面的打印尺寸

(1) 打开一个已存在的 HTML 页面，图 3-10 所示为本实例中使用的页面。

(2) 添加 CSS 样式，首先利用:first 伪类来设置打印时第一页的样式。这里为外间距 4 厘米，代码如下所示：

```
@page :first {
margin:4cm;
}
```

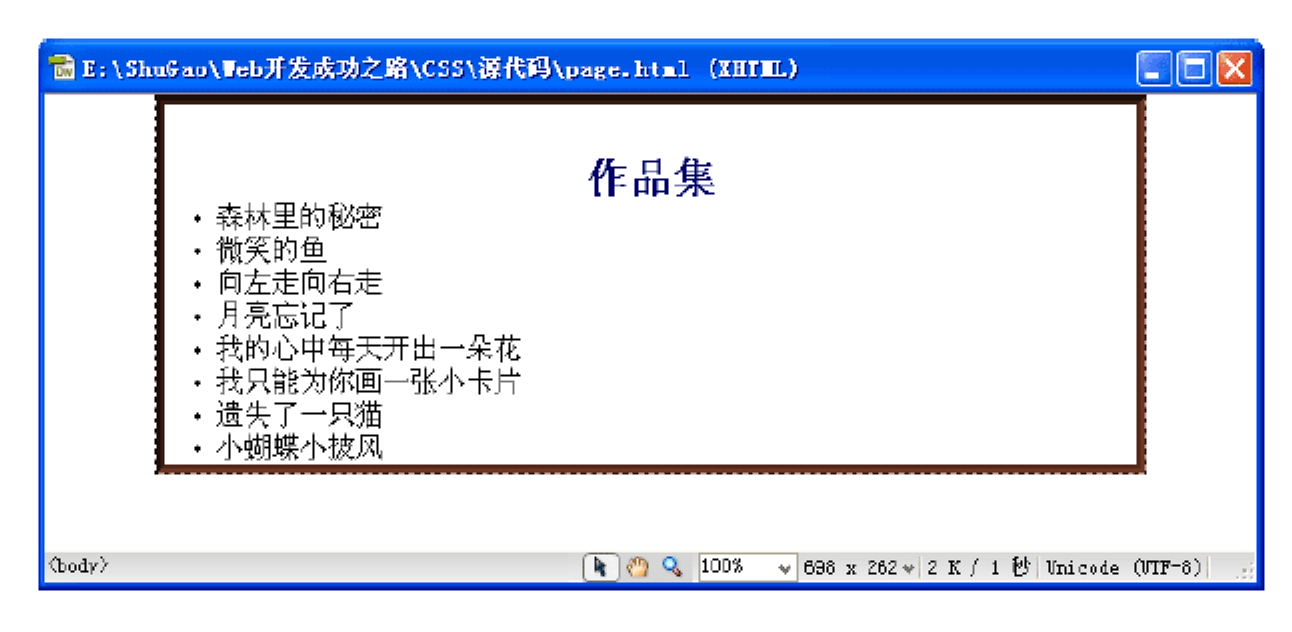


图 3-10　打开要设置的 HTML 页面

(3) 定义页面打印时左边的样式：

```
@page :left {
margin:4cm;
}
```

(4) 定义页面打印时右边的样式：

```
@page :right {
margin: 4cm;
}
```

(5) 保存对 HTML 页面的修改，完成实例的制作。

3.4.4　运行结果

在浏览器中打开已设置打印样式的页面，然后右击选择【打印预览】命令查看打印预览后的效果，也可以调整样式的值观察变化。

图 3-11 所示为未应用样式前的打印预览效果，添加样式之后的效果如图 3-12 所示。

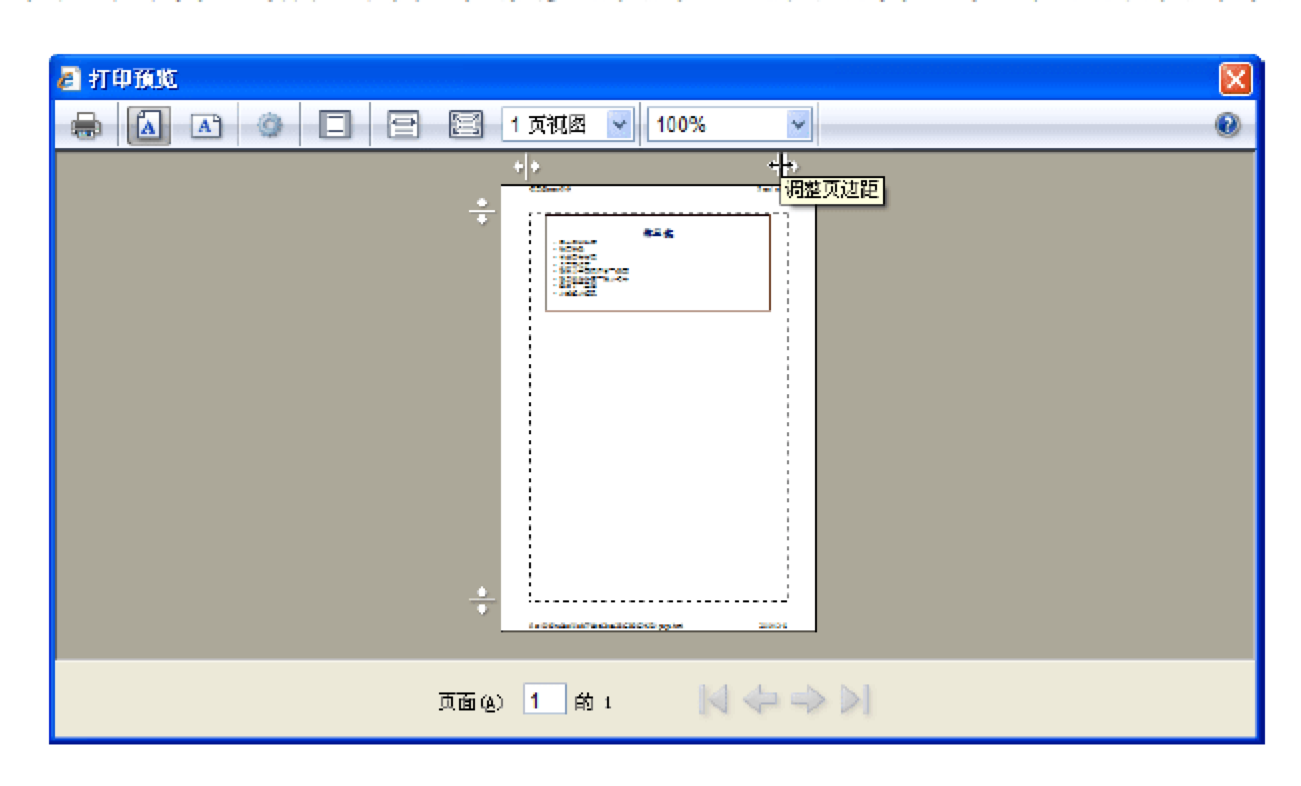


图 3-11　未应用样式前的打印预览效果

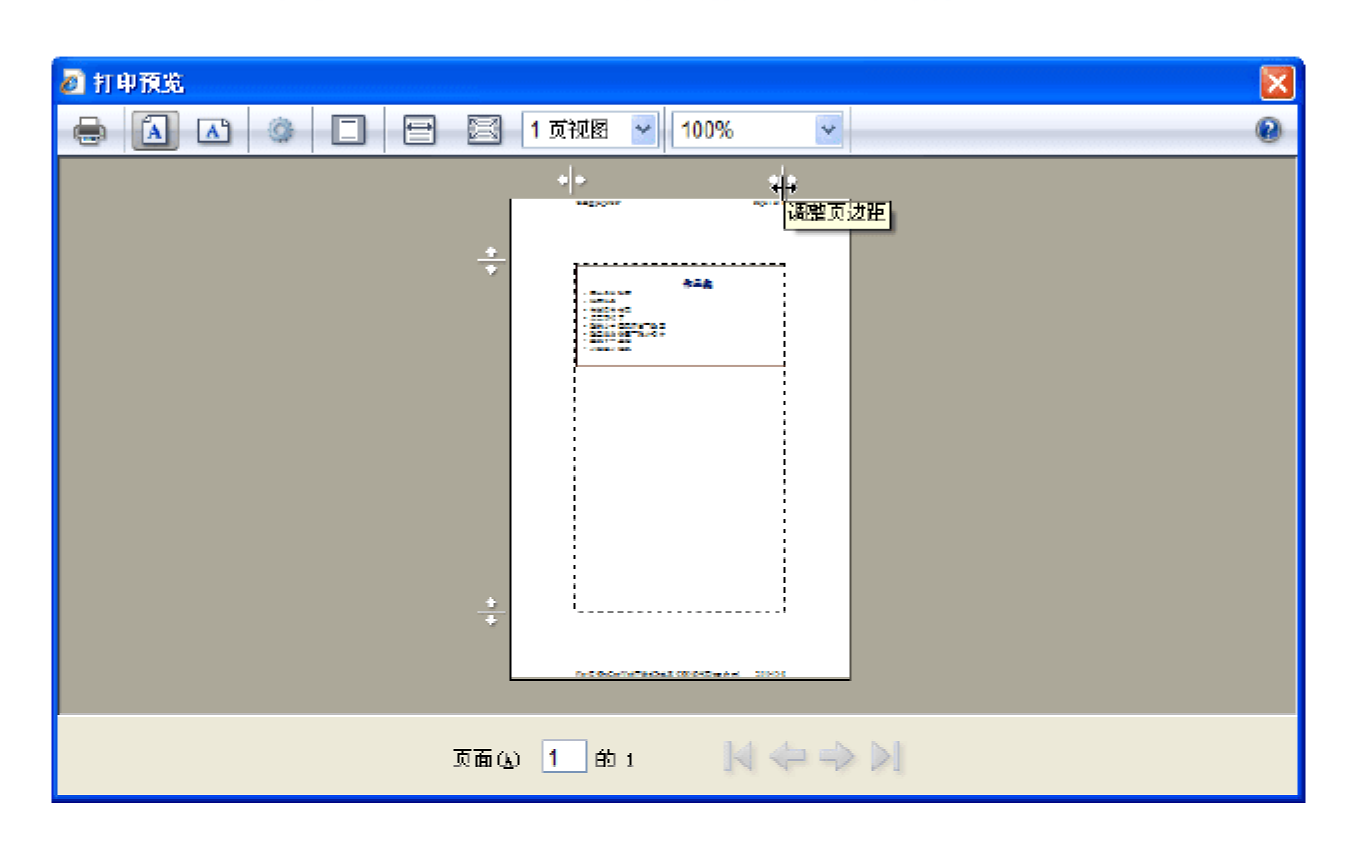


图 3-12　应用样式后的打印预览效果

3.4.5　实例分析



源码解析：

对比图 3-11 和图 3-12 的运行效果，可以看到在未应用打印样式之前，页面可以完整地预览，而且顶部、左边和右边的间距都相同。

应用打印样式之后，页面的间距增大，致使页面的内容被挤出打印范围，显示不完整了。

总的来说，:first 伪类、:right 伪类和:left 伪类在实际开发时很少使用，但是在某些情况下可以巧妙地解决打印问题。

3.5 修饰文字的特殊样式

如果想把书名号(《》)修改为引号(“”)，为了能引起浏览者的注意，使引号中的内容高亮显示，该如何实现呢？

要使引号中的内容高亮显示，可以根据以前学过的“内嵌样式表”、“类选择器”、“id 选择器”来定义内容的样式。但是如果想要修改“特殊语言字符的类型”，直接进行文本修改，既耗时，又费神，有没有一种方法一举两得呢？

别急！待我们说完:lang 伪类再回来研究。

**视频教学：**光盘/视频/3/3.5 修饰文字的特殊样式.avi　　长度：5 分钟

3.5.1 基础知识——:lang 伪类

:lang 伪类用来设置使用特殊语言的内容的样式，语法格式如下：

```
Selector : lang { sRules }
```

例如，在下面的例子中，:lang 伪类为属性值为 no 的 q 元素定义引号的类型。

```
<html>
<head>
<style type="text/css">
q:lang(no)
   {
   quotes: "~" "~"
   }
</style>
</head>
<body>
<p>文字<q lang="no">段落中的引用的文字</q>文字</p>
</body>
</html>
```

执行后，在浏览器中看到如图 3-13 所示的效果。

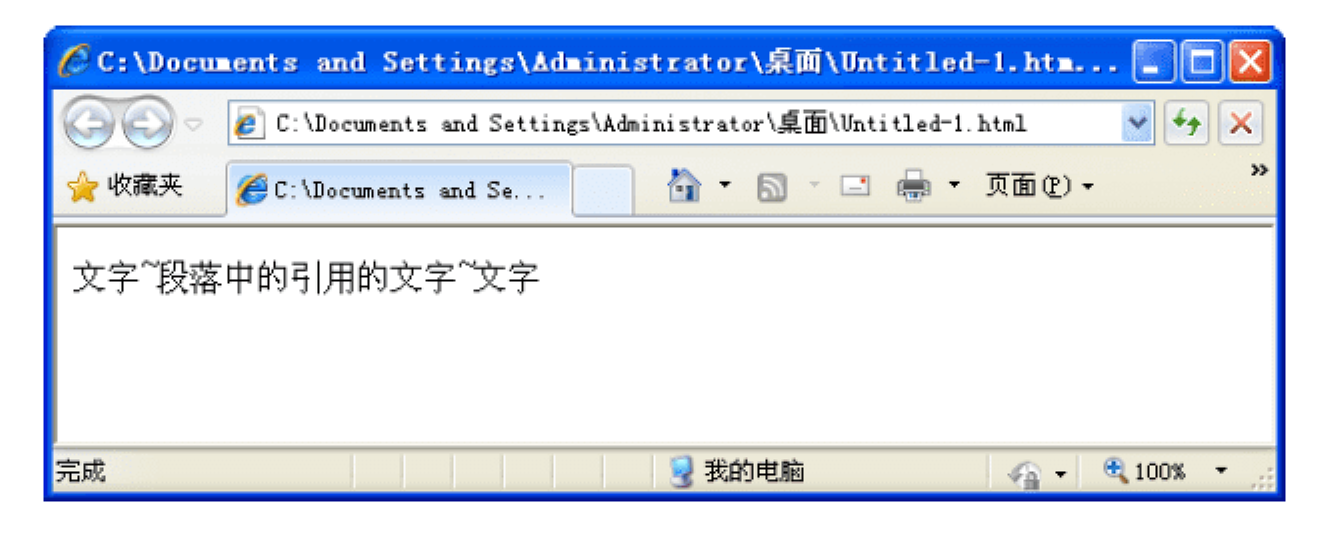


图 3-13 实例运行效果

3.5.2　实例描述

在看文章时有可能遇到类似“文章内容 ~段落中的引用~ 文章内容”这样的文本。若想要改变“~段落中的引用~”中的特殊字符的类型，但又不想在文本中直接修改，可以使用:lang伪类来巧妙地解决，只需简单地修改一下样式的类型就可以了。

3.5.3　实例应用

【例 3-5】修饰文字的特殊样式

(1) 新建一个 HTML 页面，命名为 lang.html。

(2) 在页面中添加 CSS 样式，利用:lang 伪类来修饰标题的样式。

```
q:lang(bt) {
 quotes:"----------------" "----------------"
}
```

(3) 添加用于修饰备注文本的 CSS 样式。

```
 q:lang(bz) {
 quotes:"“" "”";
 color:blue;
}
```

(4) 添加一个 CSS 样式来修饰文章中的段落。

```
q:lang(dl) {
 quotes:"[" "]"
}
```

(5) 添加一个用来修饰文章中附加信息的 CSS 样式。

```
q:lang(no) {
 quotes: "《" "》";
 color:red;
}
```

(6) 通过上面的步骤，创建了 4 个修饰样式。这一步需要在 body 元素中添加具体的文章内容并引用上面定义的样式。代码如下所示：

```
<div id="wrap">
  <p><q lang="bt">海燕</q></p>
  <p><q lang="bt">高尔基</q><q lang="bz">玛克西姆·高尔基</q></p>
  <p><q lang="dl">1</q>在苍茫的大海上，狂风卷集着乌云。在乌云和大海之间，海燕像黑色的闪
电，在高傲地飞翔。</p>
  <p><q lang="dl">2</q>一会儿翅膀碰着波浪，一会儿箭一般地直冲向乌云，它叫喊着——就在这
乌儿勇敢的叫喊声里，乌云听出了欢乐。</p>
```

```
  <p><q lang="dl">3</q>在这叫喊声里——充满着对暴风雨的渴望！在这叫喊声里，乌云听出了愤
怒的力量、热情的火焰和胜利的信心。</p>
  <p>。。。。。。后面省略</p>
  <p>作者代表作品<q lang="no">海燕之歌</q>、<q lang="no">童年</q>、<q lang="no">
在人间</q>、<q lang="no">我的大学</q> </p>
</div>
```

(7) 保存对 HTML 页面的修改，完成实例的制作。

3.5.4 运行结果

在浏览器中打开 lang.html 文件查看运行效果，如图 3-14 所示。



图 3-14 运行效果

3.5.5 实例分析



源码解析：

通过本实例，我们看到在不修改 HTML 页面内容结构的情况下，实现对特定的文字显示特殊样式的效果。

在使用:lang 伪类时，q 元素表示要定义的引用类型，在这里可以改变字符的呈现样式。

但是要注意：只有在指定了!DOCTYPE 元素后，才能在 IE8 或者更高版本里看到这个伪类起的作用。也就是说，这个伪类在 IE7 或者更早的浏览器中是看不到效果的。

3.6 个性化的歌词显示页

一些报纸或者杂志为了达到版面上某个栏目突出显示的效果，会有意地将第一个字的字号增大一倍或者将第一行文本的显示与其他行文本显示设置成不同的颜色。

在网页上如果要实现这种效果可以利用:first-letter 伪元素和:first-line 伪元素。

**视频教学：**光盘/视频/3/3.6　个性化的歌词显示页.avi　　长度：5 分钟

3.6.1　基础知识——:first-letter 伪元素和:first-line 伪元素

:first-letter 伪元素用于向文本的第一个字母添加特殊样式，语法格式如下：

```
Selector : first-letter { sRules }
```

例如：

```
p:first-letter
  {
  color:#ff0000;
  font-size:xx-large;
  }
```

:first-line 伪元素用于向某个选择器中的文字的首行添加特殊样式，语法格式如下：

```
Selector : first-line { sRules }
```

例如：

```
p:first-line
{
    color:blue;
    font-size:18px;
}
```

3.6.2　实例描述

前面介绍了伪元素 first-letter 和 first-line 的知识，是不是感觉很简单啊。下面通过制作个性化的歌词显示页面看看它们的具体应用，其中使用 first-line 强调歌曲名称，使用 first-letter 突出显示歌词的第一个字。

3.6.3　实例应用

【例 3-6】个性化的歌词显示页

(1) 新建一个 HTML 页面，命名为 ningxia.html。

(2) 在页面中添加 CSS 样式。首先利用:first-letter 伪类定义一个修饰首字符的样式，代码如下所示：

```
.text:first-letter {
    color:#FFF;
    font-size:36px;
```

```
    font-weight: bold;
    font-style: italic;
    text-decoration: underline;
}
```

(3) 再利用:first-line 伪类来定义一个修饰首行的样式，代码如下所示：

```
.title:first-line {
    margin:0px;
    padding:0px;
    color:#006;
    font-size:24px;
    font-weight:bold;
    text-align:center;
}
```

(4) 接下来在 body 元素中添加与歌曲相关的内容，包含歌曲名称、歌手、所属专辑和歌词正文，并引用上面定义的样式。代码如下所示：

```
<div id="wrap">
  <p class="title">歌曲：宁夏<br />
    专辑：爱的大游行 live 全纪录</p>
  <p class="text">宁静的夏天<br />
    天空中繁星点点<br />
    心里头有些思念<br />
    思念着你的脸<br />
    我可以假装看不见<br />
    也可以偷偷地想念<br />
    直到让我摸到你那温暖的脸<br />
    知了也睡了<br />
    安心地睡了<br />
    在我心里面宁静的夏天<br />
    那是个宁静的夏天<br />
    你来到宁夏的那一天<br />
  </p>
</div>
```

(5) 保存对 HTML 页面的修改，完成实例的制作。

3.6.4 运行结果

在浏览器中打开 ningxia.html 文件查看运行效果，如图 3-15 所示。



图 3-15　运行效果

3.6.5　实例分析


源码解析：

通过本实例，我们看到:first-letter 伪元素和:first-line 伪元素的使用并不复杂，只要分清这两个伪元素的作用范围就可以了。

最后需要注意的是：:first-letter 伪元素和:first-line 伪元素都只能作用于块对象。内联元素要使用该属性，必须先设定对象的 height 或 width 属性，或者设定 position 属性为 absolute，或者设定 display 属性为 block。

3.7　替换指定位置

大家都知道 before 和 after 是前、后的意思。但是奇怪的是，CSS 中的:before 伪元素和:after 伪元素是为源文档中不存在的内容设置样式的。

没有内容怎么设置样式呢？别急！它们有一个 content 属性，一起使用就可以为某个选择器前、后的内容设置样式了。

下面就来了解一下:before 伪元素和:after 伪元素的用法。


视频教学：光盘/视频/3/3.7　替换指定位置.avi　　　　长度：7 分钟

3.7.1　基础知识——:before 伪元素和:after 伪元素

:before 用来和 content 属性一起使用，设置在对象之前(依据对象树的逻辑结构)显示的内容，

语法格式如下：

```
Selector : before { sRules }
```

例如：

```
q:before {
 content: open-quote;
 color: red
 }
```

:after 用来和 content 属性一起使用，设置在对象之后(依据对象树的逻辑结构)显示的内容，语法格式如下：

```
Selector : after { sRules }
```

例如：

```
body:after {
   content: "The End";
   display: block;
   margin-top: 2em;
   text-align: center;
 }
```

从语句中可以看到:before 和:after 都需要与 content 属性一起使用。在表 3-1 中列出了常用 content 参数名称及说明。

表 3-1　常用 content 参数表

参数名称	说　明
attr(alt) :	使用 alt 特性的文字
counter(name)	使用已命名的计数器
counter(name, list-style-type)	使用已命名的计数器并遵从指定的 list-style-type 属性
counters(name, string)	使用所有已命名的计数器
counters(name, string, list-style-type)	使用所有已命名的计数器并遵从指定的 list-style-type 属性
no-close-quote	不插入 quotes 属性的后元素。但增加其嵌套级
no-open-quote	不插入 quotes 属性的前元素。但减少其嵌套级别
close-quote	插入 quotes 属性的后元素
open-quote	插入 quotes 属性的前元素
string	使用引号括起的字符串
url	使用指定的绝对或相对地址

假设，有一个 HTML 页面中包含了如下的代码：

```
<style type="text/css">
h1:before{ content: "标题: ";   }
p{color:blue}
</style>
```

```
<h1>花开已远</h1>
<p>你所去的地方，是不是没有寒冷，也没有忧伤。</p>
```

运行后会发现，在“花开已远”文字之前多出了“标题：”文字，效果如图 3-16 所示。

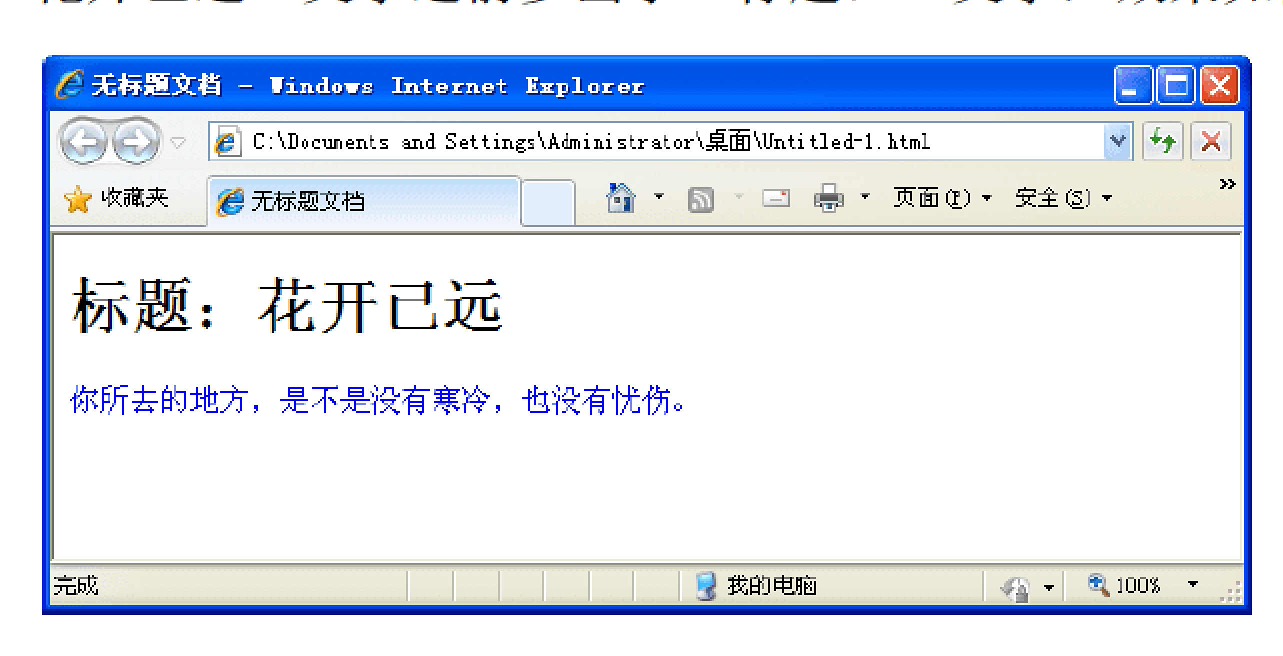


图 3-16　实例运行效果

3.7.2　实例描述

我们都熟悉 HTML 页面的运行机制吧，是类似于树状的逻辑结构，从上到下依次执行。而使用 CSS 中的:before 伪元素和:after 伪元素可以在指定元素的前面或者后面插入内容。

3.7.3　实例应用

【例 3-7】替换指定位置

(1) 新建一个 HTML 页面，命名为 zuopin.html。

(2) 在页面的 body 元素中添加如下代码：

```
<div id="wrap">
  <h1>作品集</h1>
</div>
```

(3) 使用:before 伪元素在 h1 元素之前显示一个“《”字符：

```
h1:before {
    content: "《";
}
```

(4) 使用:after 伪元素在 h1 元素之后显示一个“》”字符：

```
h1:after {
    content: "》";
}
```

(5) 接下来在 h1 元素之后添加一个 ul 列表，包含的代码如下所示：

```
<ul>
  <li>森林里的秘密</li>
  <li>微笑的鱼</li>
  <li>向左走向右走</li>
```

```
  <li>月亮忘记了</li>
  <li>我的心中每天开出一朵花</li>
  <li>我只能为你画一张小卡片</li>
  <li>遗失了一只猫</li>
  <li>小蝴蝶小披风</li>
</ul>
```

(6) 根据第(3)和第(4)步的方法，在这里为 li 元素的前后添加"“"和"”"符号，并以红色显示。这部分代码如下所示：

```
li:before {
    content: open-quote;
    color:#F00;
}
li:after {
    content:close-quote;
    color:#F00;
}
```

(7) 最后，在 body 元素之后添加一个"The End"作为结束字符串。

```
body:after {
    content: "The End";
    display: block;
    margin-top: 10px;
    text-align: center;
}
```

(8) 保存对 HTML 页面的修改，完成实例的制作。

3.7.4 运行结果

在浏览器中打开 zuopin.html 文件查看运行效果，如图 3-17 所示。

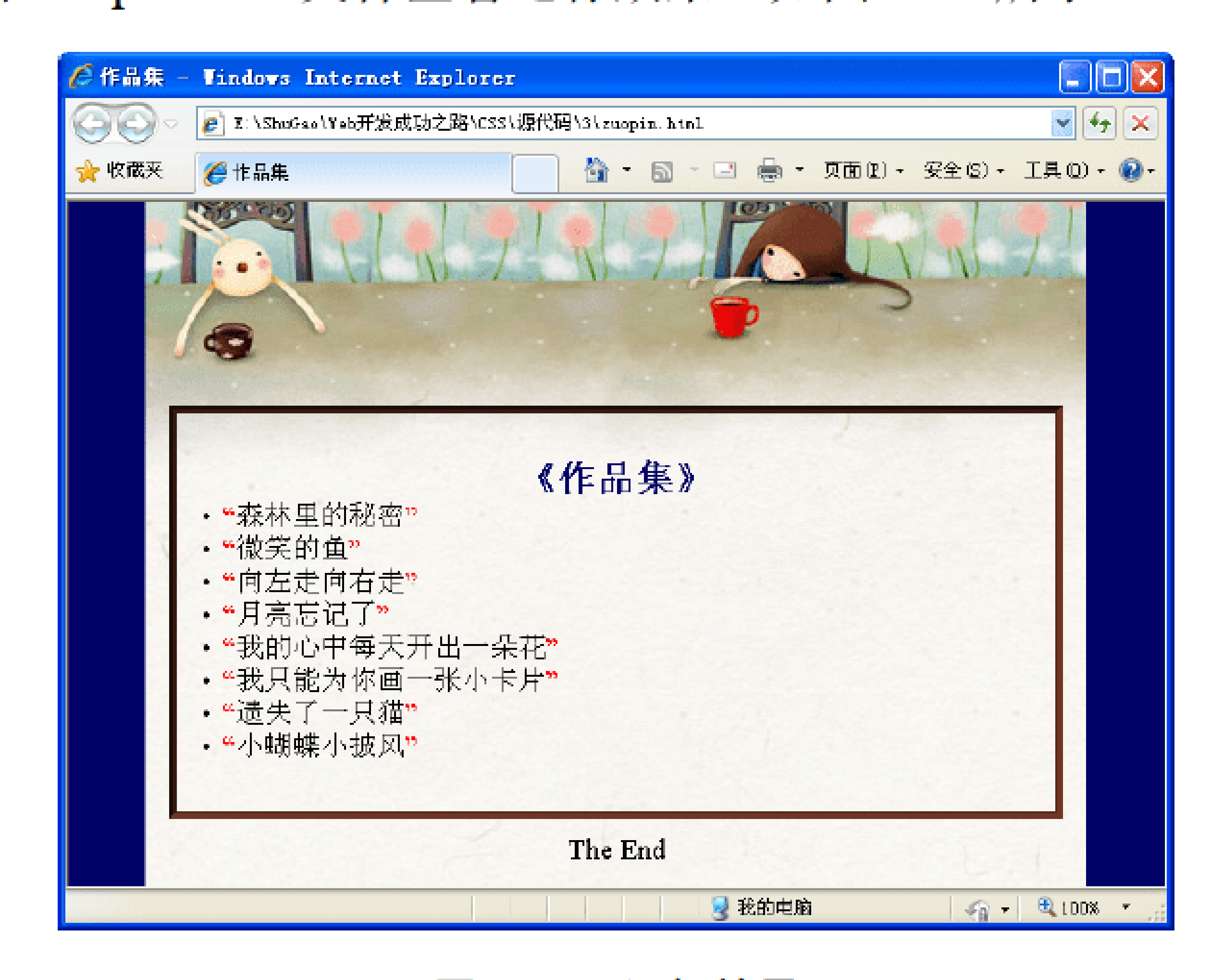


图 3-17 运行效果

3.7.5　实例分析



源码解析：

通过本实例，我们真正掌握了:before 伪元素和:after 伪元素的使用方法。它们的唯一区别就是：一个是定义对象之前显示的内容，另一个是定义对象之后显示的内容。

然后，就是 content 属性参数的设置了。它可以包含字符串，也可以是表 3-1 中给出的其他值。

3.8　常见问题解答

3.8.1　CSS 中伪类的应用问题



CSS 中伪类的应用问题。
网络课堂：http://bbs.itzcn.com/thread-11199-1-1.html

假设，在我的页面中有如下的 CSS 代码：

```
a:link {color: #003399;}
a:visited {color: #000066;}
a:hover {color: #000000;}
a:active {color: #000000;}
```

上面代码定义了链接的 4 种状态，它在 IE7、Firefox 浏览器下对所有块级元素都可以使用。例如：

```
li:hover  {border:1px solid #2089D8;}
```

li:hover 在 IE7、Firefox 都会产生效果，但是在 IE6 里却没有效果，有没有什么办法可以使得在 IE6 里也可以达到这个效果？

【解决办法】这是因为在 CSS 1 中此伪类仅可用于 a 对象。对于无 href 属性的 a 对象，此伪类不发生作用。而在 CSS 2 中此伪类可以应用于任何对象。

另外，IE 6 不能完全支持 CSS 2 所提供的功能。

当然也不是没有解决办法，可以添加如下样式：

```
li a:hover {border:1px solid #2089D8;display: block;text-decoration: none;}
```

然后用如下 HTML 来引用这个样式。

```
<li><a href="#">TEXT</a></li>
```

3.8.2 关于 CSS 伪类鼠标效果



关于 CSS 伪类鼠标效果。

网络课堂：http://bbs.itzcn.com/thread-11200-1-1.html

a:link 和 a:visited 这些里面可不可以添加图片而不改文字样式，做成像鼠标经过图像那种的效果？具体添加图片的代码是什么？

【解决办法】鼠标经过时样式是用伪类 a:hover，a:link 是未被访问前的样式，a:visited 是访问过后的样式。

在<HTML>和<BODY>元素之间插入如下代码：

```
<style type="text/css">
<!--
#a1 a:hover{background-image :url("鼠标滑过.jpg"); }
#a1 a:link {background-image :url("未访问前.jpg"); }
-->
</style>
```

上面的代码用于定义一个 ID 选择符。然后引用上面样式的 html 源码如下：

```
<div id="a1">
<a href="#">我是未访问前的</a>
</div>
```

3.8.3 关于 first-child 伪类无效的问题



关于 first-child 伪类无效的问题。

网络课堂：http://bbs.itzcn.com/thread-11201-1-1.html

在对表格使用 first-child 伪类时，需要给所有表格的第一行设置样式，按说可以使用下面的代码来进行设置。

```
table > tr:first-child{
  background:red;
  }
```

但是没有效果，一定要用下面的代码才可以达到想要的效果。

```
tr:first-child{
  background:red;
  }
```

我是新手，对这个伪类的用法总是似懂非懂，请高手解释一下其中的道理。

【解决办法】table > tr:first-child 之所以不能选中第一个 tr，就是因为在 table 和 tr 中有一个 tbody。虽然没有显式地写明 tbody，但是在解析表格中还是会自动解析出来的。所以，你的

代码是错误的，改用 tr:first-child 就可以了。

3.8.4　box:after{}的功能



CSS 问题.box:after{}是什么意思？
网络课堂：http://bbs.itzcn.com/thread-11202-1-1.html

在一个 CSS 模板中发现有下面两行代码：

```
.box:after{ content: "."; },
.box:after{ content:"\0020"; },
```

运行了一下页面，没看到有什么效果。请教各位，那两行代码是什么意思？

【解决办法】其实很简单啊。就是使用了 CSS 中的:after 伪类，它的含义就是在匹配的选择符之后追加的内容。

在你这两行代码中对 content 定义了两次，而后面 content 的定义会覆盖前面的定义。追加的位置是在 class 名称为 box 的元素后。

3.9　习　　题

1. 填空题

(1) 在 CSS 的链接伪类中，__________必须被置于 a:link 和 a:visited 之后才有效。

(2) 要使某个元素下第一个 li 子元素变成红色，应该使用代码__________。

(3) 假设有如下的一段 HTML 代码:

```
<ul>
  <li>新 闻</li>
  <li>网 页</li>
  <li>贴 吧</li>
  <li>知 道</li>
  <li>图 片</li>
</ul>
```

要使 li 元素中第一个字符为红色、字号为 18 像素，应该使用代码__________。

(4) 在使用伪类时，q 元素表示要定义的引用类型__________。

2. 选择题

(1) 在下面的 CSS 伪类中使用__________来定义鼠标移动到链接上的样式。

A. a:hover　　B. a:link　　C. a:visited　　D. a:active

(2) __________伪类用来设置对象在成为焦点时的样式。

A. :first-child　　B. :focus　　C. :lang　　D. :before

(3) 设置页面容器第一页使用的样式应该是__________伪类。

A. :first　　B. :first-child　　C. :first-letter　　D. :first-line

(4) 下面__________不是 content 的参数值。

A. image　　B. string　　C. url　　D. attr

(5) 在下面的伪类中__________不能控制打印尺寸。

A. :first　　B. :top　　C. :right　　D. :left

3. 上机练习

上机练习 1：实现多种链接效果

通过本章的学习，我们知道链接伪类是众多 CSS 伪类中使用频率最高的。它们是 a:link、a:visited、a:hover 和 a:active，本次上机练习的目的也是针对该伪类的。

要求读者使用链接伪类实现多种不同的链接效果。如图 3-18 所示是 a:link 效果，图 3-19 所示是 a:hover 效果，图 3-20 所示是 a:visited 效果，图 3-21 所示是 a:active 效果。



图 3-18　a:link 效果

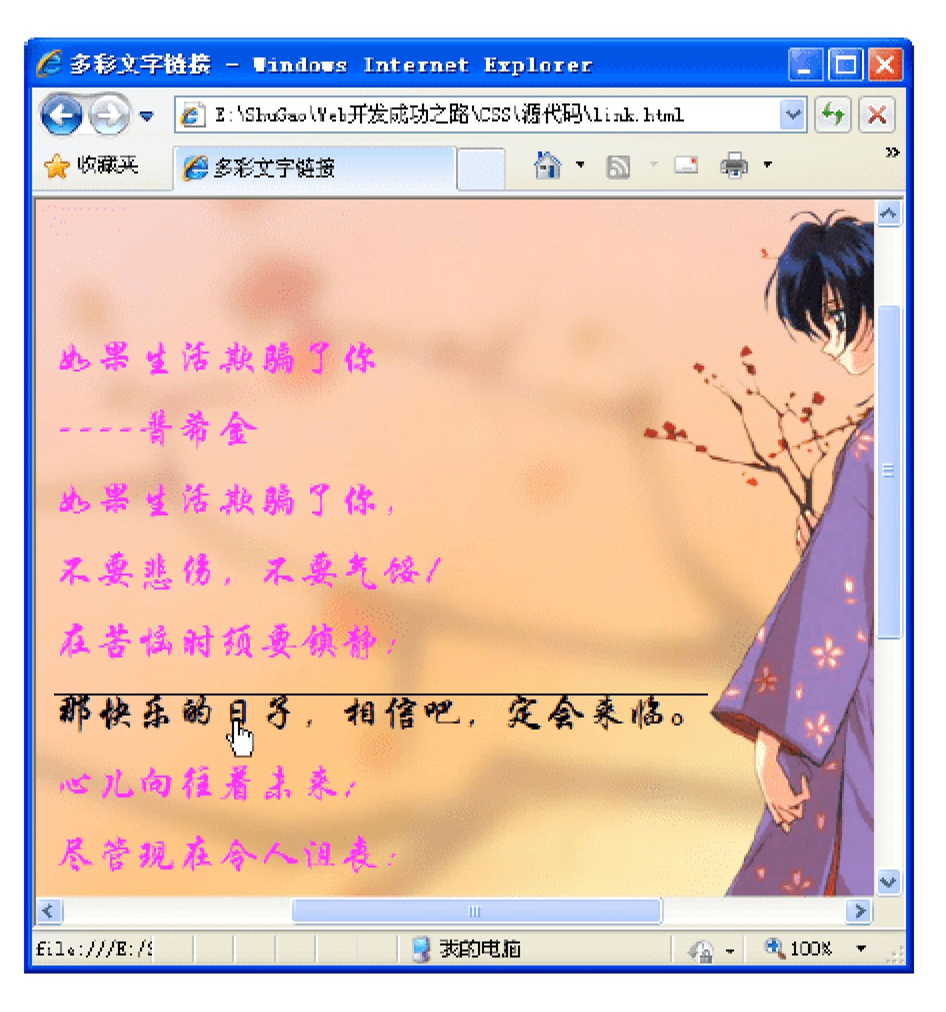


图 3-19　a:hover 效果

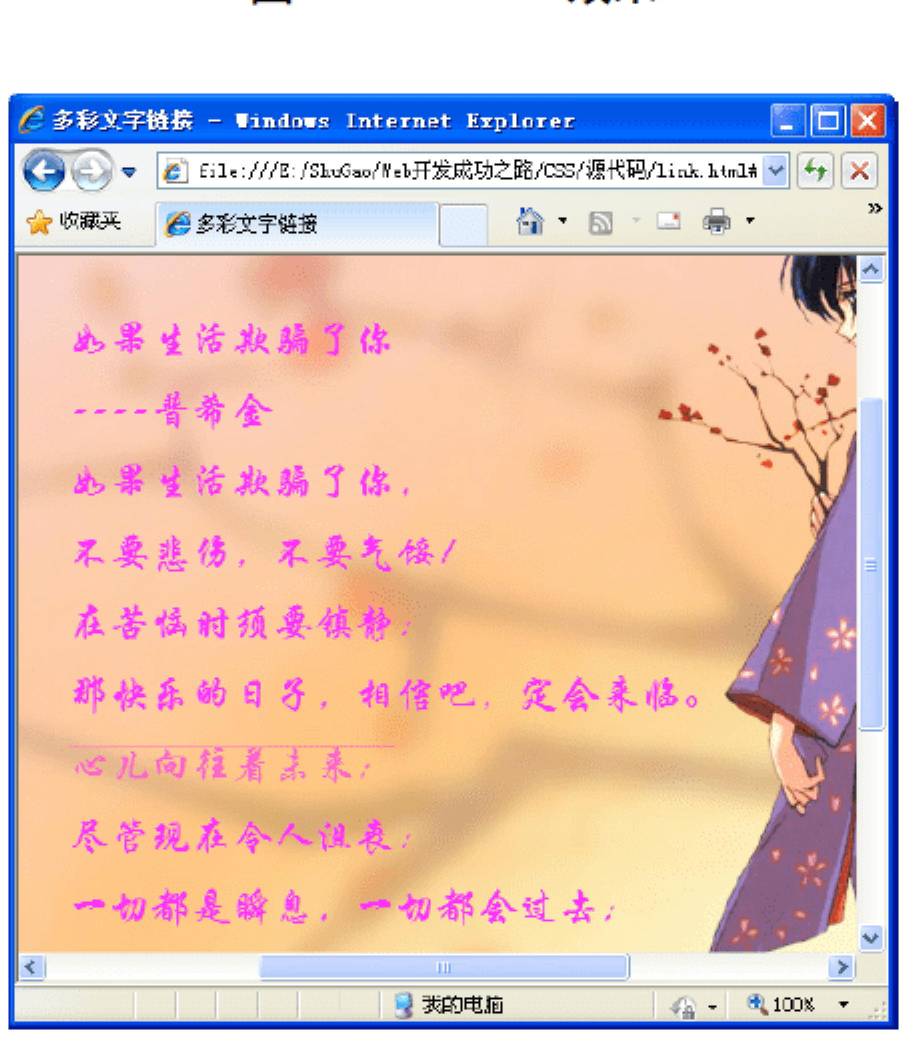


图 3-20　a:visited 效果

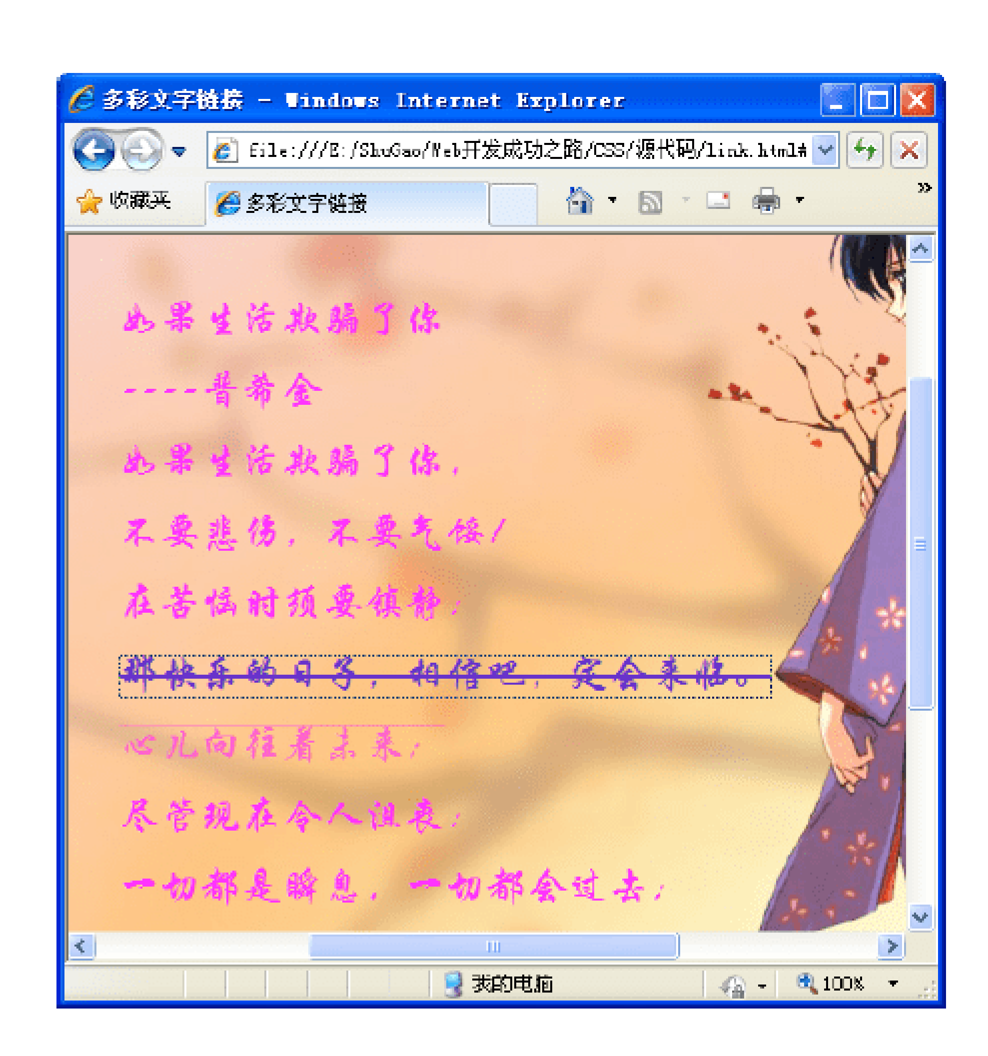


图 3-21　a:active 效果

上机练习 2：实现对 HTML 页面追加内容

假设，在一个 HTML 中的 body 元素中仅仅包含了如下的代码：

```
<div align="center" > <q>谁要在我前面显示，简直是在太岁头上动土！</q> </div>
<div align="center" > </div>
<div align="center" > </div>
```

现在要求读者在不改变 HTML 页面内容结构的情况下，向其中追加内容，实现如图 3-22 所示的运行效果。

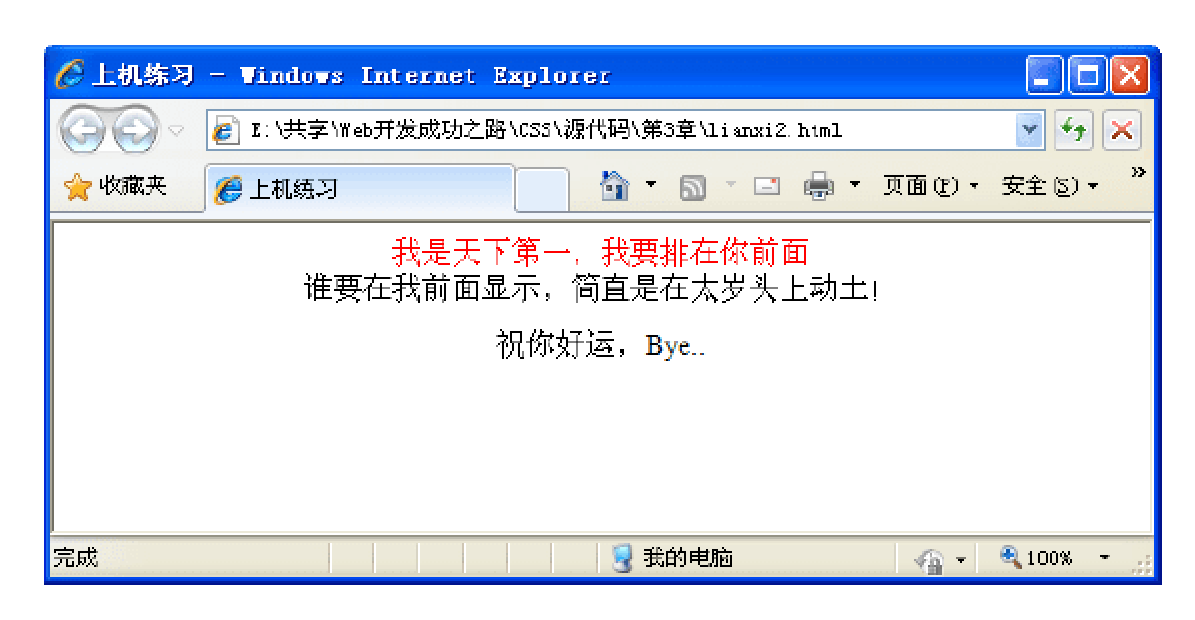
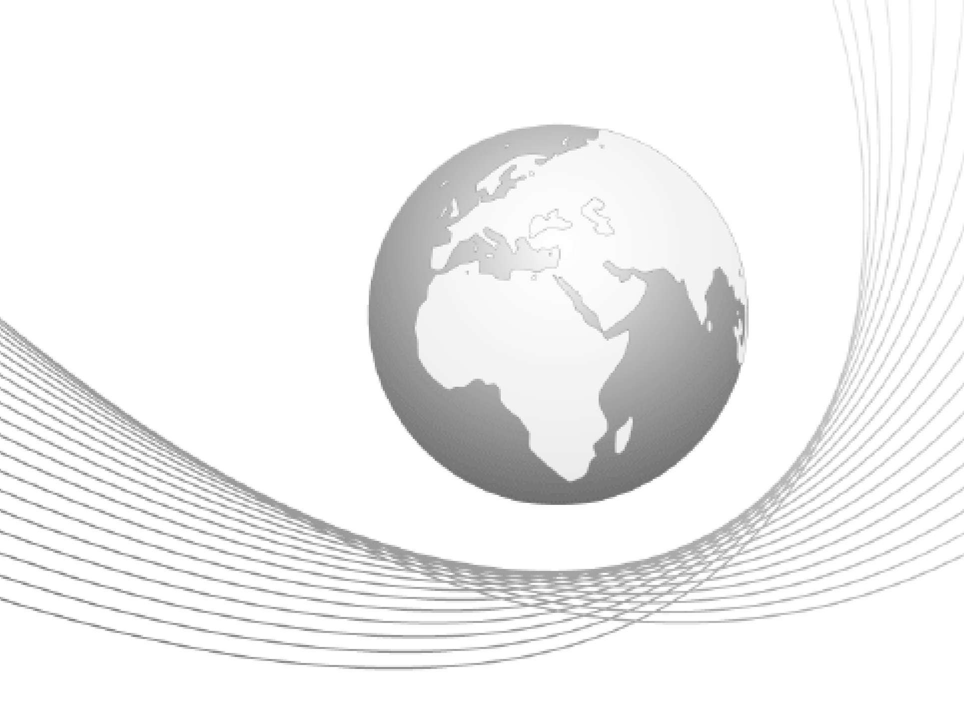


图 3-22　运行效果



第4章　字体属性

内容摘要：

网页最重要的是给客户提供信息，而信息的载体是文本，文本是由字体组成的。CSS 样式表分别提供了字体属性和文本属性来对页面进行更精确的布局。在本章中将详细介绍字体属性，文本属性将在下一章介绍。

学习目标：

- 掌握颜色属性的应用
- 掌握字体属性的应用
- 掌握字体外部修饰属性的用法
- 了解间隔属性 letter-spacing 和 word-spacing
- 了解行高属性 line-height 的用法

4.1 炫 彩 诗

当看到做出来的网页中的字体总是一种灰暗的颜色时，是否感到页面效果很单调呢？接下来将介绍 CSS 的颜色属性，它能够使页面的字体色彩变得丰富多彩。

**视频教学：**光盘/视频/4/4.1　炫彩诗.avi　　长度：5 分钟

4.1.1 基础知识——Color

颜色用于设定字体及背景的颜色显示，CSS 中共预设了 16 种颜色及它们的衍生色，这 16 种颜色是 CSS 规范推荐的，能够被大多数主流浏览器所识别，如表 4-1 所示。

表 4-1　CSS 推荐颜色

颜　色	名　称	颜　色	名　称
aqua	浅绿色	black	黑色
blue	蓝色	fuchsia	桃红色
gray	灰色	green	绿色
lime	亮绿色	maroon	栗色
navy	藏青色	olive	橄榄色
purple	紫色	red	红色
silver	银色	teal	青色
white	白色	yellow	黄色

这些颜色在代码中可以直接以名称方式使用，如以下代码：

```
div {color:red; }
```

除了这 16 种基准颜色之外，还可以使用 CSS 定义的其他预定义颜色，大多数浏览器能够识别包括这 16 种颜色在内的 140 多种颜色，例如，silver、PaleGreen 等。

此外，为了让页面色彩更加丰富，还可以使用十六进制颜色和 RGB 颜色。



用颜色名称指定 color 可能不被一些浏览器所接受。但使用 RGB 值指定颜色能被所有的浏览器识别并正确显示。

4.1.2 实例描述

大家初学 CSS，相信对 color 属性的理解还不是很透彻吧，不过不要担心，下面我们通过一个“炫彩诗”的实例来深入地学习和理解 color 属性的用法。

4.1.3　实例应用

【例 4-1】炫彩诗

(1) 首先，新建一个名为 poetry 的 html 文件，在文件中写入以下代码：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>炫彩诗</title>
</head>
<body>
   <div id="div2">
      <p class="p0" >江楼月</p>
      <p class="pz">—作者：白居易</p>
      <p class="p1">嘉陵江曲曲江池，明月虽同人别离。</p>
      <p class="p2">一宵光景潜相忆，两地阴晴远不知。</p>
      <p class="p3">谁料江边怀我夜，正当池畔望君时。</p>
      <p class="p4">今朝共语方同悔，不解多情先寄诗。</p>
   </div>
  </body></html>
```

(2) 在<head>标签内部，通过 CSS 的颜色样式来控制 div 和 p 标签的样式，以实现需要达到的效果。其代码如下：

```
<style type="text/css">
   body{
     background-color: bisque;
     background-image: url("img/bg.jpg");
   }
   #div2{
     font-size: 14pt;
     margin-top: 200px;
     margin-left: 350px
   }
   .p0{
      color: red;
      margin-left: 110px
   }
   .pz{
      font-size:12pt;
      margin-left: 150px
   }
   .p1{color: blue}
   .p2{color: RGB(150,65,220)}
   .p3{color: #00FF00}
   .p4{color: saddlebrown}
   </style>
```

4.1.4 运行结果

下面通过 IE 浏览器打开 poetry.html 文件，就可以看到以下效果，如图 4-1 所示。



图 4-1 炫彩诗效果

4.1.5 实例分析



源码解析：

上述实例创建了一个静态页面，通过<div>标签来控制布局，在<p>标签的伪类中定义 color 属性，用来显示所需要的效果，最终得到如图 4-1 所示的绚丽多彩的页面效果。

在不同的操作系统或浏览器中，同一种颜色的显示可能会不同。所以要尽量使用网络安全色。例如，RGB 值为 51 倍数的颜色值，十六进制值为 00、33、66、99、CC 和 FF 的组合，都是安全的。

4.2 编 排 网 页

每个人在浏览页面时，对页面的第一印象可能就是文档所使用的字体和页面上的各种颜色。一个页面字体使用的好坏可以直接关系到页面的浏览质量。因此，在 CSS 中有关字体的属性都具有非常强大的功能。下面就来看看有关字体的属性吧。

**视频教学：**光盘/视频/4/4.2 编排网页.avi　　长度：8 分钟

4.2.1 基础知识——字体

CSS 字体(font)属性用来定义文本中的字体。设置字体属性是样式表的常见用途之一。CSS 字体属性允许设置字体系列(font-family)和字体加粗(font-weight)，还可以设置字体的大小、字

体风格(如斜体)和字体变形(如小型大写字母)等。

1. font-style

font-style 属性用于定义字体显示的样式，有三种可选属性值：normal、italic 和 oblique。其中 normal 是默认值，表示正常的字体样式；italic 表示将文本设置为斜体样式；oblique 表示将没有斜体变量的特殊字体设置为斜体样式。例句如下：

```
.s1{ font-sytle:italic}
```

下面通过一个小实例来演示 font-style 的三种属性值的用法，实例部分代码如下：

```
<title>字体风格属性 font-style</title>
<STYLE>
.box_grey p{
        font-style:italic
        }
    .box_grey h3{
       font-style:oblique
    }
</STYLE>
…
<div class="box_grey">
                <h3><img src="images/page_notice.gif" width="19px"
height="16px" alt="textbox" /> <strong>蓝色心情许愿墙</strong></h3>
                <p> 开夜的天窗，对着划过的流星许下永恒的祝愿，不变的诺言。在每个人的心底，
都有最美好的祝愿。或许，这就是一种缘份。<br />
<br />
                让我们用最虔诚的心，在这里许下您的心愿、祝福，留下您最想对他(她)说的话，
永久地挂在许愿墙上，愿望就会很快实现的哟。<br />
                </p><p>
                    每个人心中都有美好的愿望，并且渴望实现，不妨为自己或亲友写个愿望条，永
久地挂在蓝色心情许愿墙上，愿望就会很快实现的哟。蓝色心情许愿墙是互联网上第一个真正的许愿墙！
为您提供许愿及祈祷平台，致力于实现每个人心中的美好梦想，蓝色心情祝您早日实现您的美好愿望。

    许愿墙的故事：相传在几万年前，一颗鬼灵精怪的魔豆已在主的身边陪伴了五千年。他是潘多拉魔盒
里最后一件礼物——希望。主把他放在御座上的珀西盒里，护佑着天堂里的一切。在魔豆存在的地方总
是充满希望的光芒，在他面前许下的愿望都能实现。
                </p>
            </div>
```

在<style>标签里对<p>和<h3>标签使用 font-style 属性，演示结果如图 4-2 所示。

2. font-variant

font-variant 属性提供了字体的变形功能，该属性有两个属性值：normal 和 small-caps。normal 是默认值，用来描述普通文本，small-caps 表示使用小型的大写字母文本。例句如下：

```
.s1 {font-variant:small-caps}
```



图 4-2　设置 font-style 属性的实例运行效果

下面通过一个小实例来演示 font-variant 的两种属性的用法，实例中 CSS 定义的部分代码如下：

```
<html>
<head>
<title>字体变量属性 font-variant</title>
<STYLE>
.box_grey p{
        font-variant:small-caps;
        }
</STYLE>
</head>
<body>
<div class="box_grey">
                <h3><img src="images/page_notice.gif" width="19px"
height="16px" alt="textbox" /> <strong>蓝色心情许愿墙</strong></h3>
                <p> 开夜的天窗，对着划过的流星许下永恒的祝愿，不变的诺言。在每个人的心底，
都有最美好的祝愿。或许，这就是一种缘份。<br />
<br />
                让我们用最虔诚的心，在这里许下您的心愿、祝福，留下您最想对他(她)说的话，
永久地挂在许愿墙上，愿望就会很快实现的哟。<br />
                </p><p>
                    每个人心中都有美好的愿望，并且渴望实现，不妨为自己或亲友写个愿望条，永
久地挂在蓝色心情许愿墙上，愿望就会很快实现的哟。蓝色心情许愿墙是互联网上第一个真正的许愿墙！
为您提供许愿及祈祷平台，致力于实现每个人心中的美好梦想，蓝色心情祝您早日实现您的美好愿望。

    许愿墙的故事：相传在几万年前，一颗鬼灵精怪的魔豆已在主的身边陪伴了五千年。他是潘多拉魔盒
里最后一件礼物——希望。主把他放在御座上的珀西盒里，护佑着天堂里的一切。在魔豆存在的地方总
是充满希望的光芒，在他面前许下的愿望都能实现。
                </p>
            </div>
</body>
</html>
```

以上代码在浏览器中运行的效果如图 4-3 所示。



图 4-3　设置 font-variant 属性的实例运行效果

3. font-weight

font-weight 属性用于对显示元素中的文本字体进行加粗。该属性有 13 个有效值，分别是 bold、bolder、lighter、normal、100、200、300、400、500、600、700、800、900。如果没有设置该属性，则使用默认值 normal。设置值为 100～900，值越大，加粗的程度越高。其中，数字值 400 相当于关键字 normal，700 相当于 bold。表 4-2 所示列出了相关信息。

表 4-2　font-weight 属性的取值与描述

值	描　述
bold	定义粗体字体
bolder	定义更粗的字体，相对值
lighter	定义更细的字体，相对值
normal	默认值，定义标准的字体
100～900(100 的整数倍)	定义由细到粗的字体，其中，400 等同于 normal，700 等同于 bold
inherit	规定应该从父元素继承字体的粗细


任何版本的 Internet Explorer(包括 IE 8)都不支持属性值 inherit。

例句如下：

```
<p style = "font-weight:bold">这段文字字体的浓淡属性(font-weight)值是 bold。<p>
```

下面通过一个小实例来演示 font-weight 的各种属性值的用法，实例代码如下：

```
<p style = "font-weight:400"> 开夜的天窗，对着划过的流星许下永恒的祝愿，不变的诺言。
在每个人的心底，都有最美好的祝愿。或许，这就是一种缘份。<br />
<br />
                让我们用最虔诚的心，在这里许下您的心愿、祝福，留下您最想对他(她)说的话，
永久地挂在许愿墙上，愿望就会很快实现的呦。<br />
```

```
          </p>
        <p style="font-weight:bolder">
            每个人心中都有美好的愿望，并且渴望实现，不妨为自己或亲友写个愿望条，永久地挂在蓝色心情许愿墙上，愿望就会很快实现的呦。蓝色心情许愿墙是互联网上第一个真正的许愿墙！为您提供许愿及祈祷平台，致力于实现每个人心中的美好梦想，蓝色心情祝您早日实现您的美好愿望。

    许愿墙的故事:相传在几万年前，一颗鬼灵精怪的魔豆已在主的身边陪伴了五千年。他是潘多拉魔盒里最后一件礼物——希望。主把他放在御座上的珀西盒里，护佑着天堂里的一切。在魔豆存在的地方总是充满希望的光芒，在他面前许下的愿望都能实现。
          </p>
```

以上代码在浏览器中运行的效果如图 4-4 所示。



图 4-4 设置 font-weight 属性的实例运行效果

4. font-size

font-size 属性用于设置字体的大小，CSS 为该属性提供了多种设置字体值的方法，其取值如表 4-3 所示。

表 4-3 font-size 属性的取值与描述

字 体 值	描 述
xx-small x-small small medium large x-large xx-large	把字体的尺寸设置为不同的尺寸，从 xx-small 到 xx-large 默认值：medium
smaller	把 font-size 设置为比父元素更小的尺寸
larger	把 font-size 设置为比父元素更大的尺寸
length	把 font-size 设置为一个固定的值

续表

字 体 值	描　述
%	把 font-size 设置为基于父元素的一个百分比值
inherit	规定应该从父元素继承字体尺寸

在这些字体值中，xx-small、x-small、small、medium、large、x-large、xx-large 设置字体的绝对尺寸，根据对象字体进行调整，但没有具体的值进行调整，xx-large 表示最大，xx-small 表示最小。例句如下：

```
<p style = "font-size:xx-small">字体大小属性(font-size)值是 xx-small。<p>
```

下面通过一个小实例来演示 font-size 的各种属性值的用法，实例部分代码如下：

```
<p style="font-size:20px"> 开夜的天窗，对着划过的流星许下永恒的祝愿，不变的诺言。在每个人的心底，都有最美好的祝愿。或许，这就是一种缘份。<br />
<br />
                 让我们用最虔诚的心，在这里许下您的心愿、祝福，留下您最想对他(她)说的话，永久地挂在许愿墙上，愿望就会很快实现的哟。<br />
                 </p>
              <p style="font-size:12px">
                      每个人心中都有美好的愿望，并且渴望实现，不妨为自己或亲友写个愿望条，永久地挂在蓝色心情许愿墙上，愿望就会很快实现的哟。蓝色心情许愿墙是互联网上第一个真正的许愿墙！为您提供许愿及祈祷平台，致力于实现每个人心中的美好梦想，蓝色心情祝您早日实现您的美好愿望。

    许愿墙的故事：相传在几万年前，一颗鬼灵精怪的魔豆已在主的身边陪伴了五千年。他是潘多拉魔盒里最后一件礼物——希望。主把他放在御座上的珀西盒里，护佑着天堂里的一切。在魔豆存在的地方总是充满希望的光芒，在他面前许下的愿望都能实现。
                 </p>
```

在浏览器中的运行效果如图 4-5 所示。



图 4-5　设置 font-size 属性的实例运行效果

5. font-family

font-family 属性用于指定可替换的字体类型名称，如 Arial、Tahoma、Courier、华文行楷、

宋体等字体类型。例如：

```
p{
   font-family: 宋体;
}
```

font-family 可以把多个字体名称作为一个“回退”系统来保存。如果浏览器不支持第一个字体，则会尝试下一个。也就是说浏览器会使用它支持和识别的第一个值。例如：

```
p{
   font-family: "Tahoma","Times New Roman",宋体;
}
```

如果浏览器不能正确显示 Tahoma 字体，那么将以 Times New Roman 的形式显示，如果 Times New Roman 字体也不能在浏览器中显示，则使用宋体。将以上代码应用到页面中，在浏览器中运行，如图 4-6 所示。



图 4-6　设置 font-family 属性的实例运行效果

如果 font-family 属性值中的字体类型由多个字符串或空格组成，则应使用双引号。但如果在页面中使用 style 属性定义样式表，那么就要使用单引号，例如以下部分代码：

```
<p  style="font-family: 'Times New Roman'"> 开夜的天窗，对着划过的流星许下永恒的祝愿，不变的诺言。在每个人的心底，都有最美好的祝愿。或许，这就是一种缘份。<br />
<br />
                让我们用最虔诚的心，在这里许下您的心愿、祝福，留下您最想对他(她)说的话，永久地挂在许愿墙上，愿望就会很快实现的哟。<br />
                </p>
             <p style="font-family:'新宋体'">
                    每个人心中都有美好的愿望，并且渴望实现，不妨为自己或亲友写个愿望条，永久地挂在蓝色心情许愿墙上，愿望就会很快实现的哟。蓝色心情许愿墙是互联网上第一个真正的许愿墙！为您提供许愿及祈祷平台，致力于实现每个人心中的美好梦想，蓝色心情祝您早日实现您的美好愿望。

    许愿墙的故事:相传在几万年前，一颗鬼灵精怪的魔豆已在主的身边陪伴了五千年。他是潘多拉魔盒里最后一件礼物——希望。主把他放在御座上的珀西盒里，护佑着天堂里的一切。在魔豆存在的地方总是充满希望的光芒，在他面前许下的愿望都能实现。
                </p>
```

在浏览器中运行的效果如图 4-7 所示。

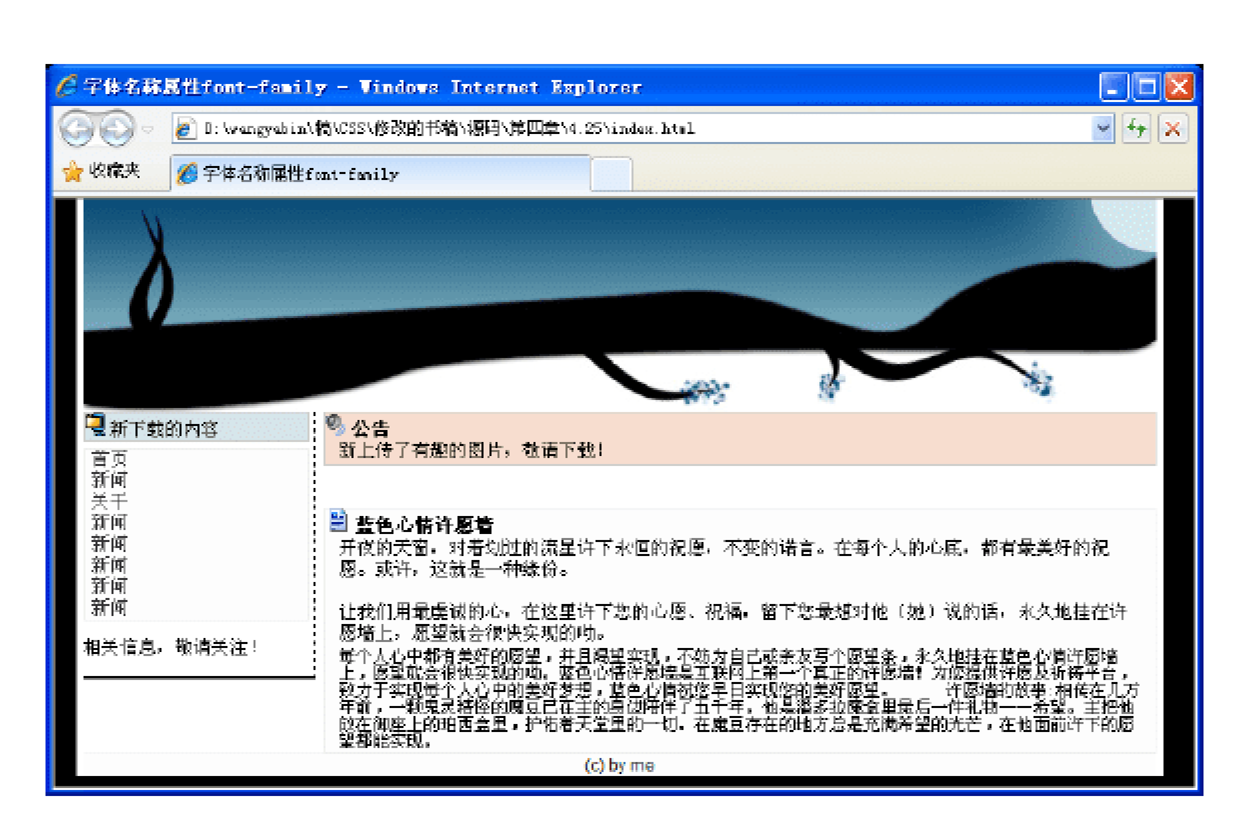


图 4-7　字体名称属性的实例运行效果

6. font

我们在设计页面时，不可能对字体只设置一种属性，而是设置多种属性，但是这些属性逐个设置起来比较麻烦，CSS 样式表提供了 font 属性来解决这一问题。

font 属性是各种字体属性的一种快捷的综合写法。这个属性可以综合字体风格属性(font-style)、字体变量属性(font-variant)、字体浓淡属性(font-weight)、字体大小属性(font-size)和字体名称属性等属性。如下所示：

```
.s2 {font:normal small-caps normal 14pt Courier}
```

font 属性内的属性值的排列顺序是 font-style、font-variant、font-weight、font-size 和 font-family，各个属性值之间要用空格隔开。但如果 font-family 属性有多个值，则使用逗号隔开。其中 font-size 和 font-family 的位置必须固定，不能改变，否则将有可能被浏览器忽略，如图 4-8 和图 4-9 所示，其中图 4-9 是 font-size 和 font-family 位置调换的效果。



图 4-8　font 属性效果



图 4-9　font 属性值排列调换的效果

4.2.2　实例描述

我们在设计网页时，是否经常遇到一个页面中会出现大小、风格等不一致的字体呢？自己动手去调试的话，有些繁杂。下面通过一个实例来介绍如何使用 CSS 字体属性编排网页。

4.2.3 实例应用

【例 4-2】编排网页

在这个实例中，我们将用到字体属性中的 font-style、font-variant、font-weight、font-size、font-family 和 font 综合属性来对页面进行编排，使我们的页面都有一个统一的风格。

目前通过 CSS 2 技术还不能直接实现普通段落文字分栏排版，要实现分栏的效果，需要借助多个 div 浮动定位来完成，下面我们将一步步实现这个实例。

(1) div 布局页面。为了制作三列的排版效果，我们将三个 div 放入#div_1 内，通过后面将要讲到的伪类.text 来控制属性，代码如下：

```
<div class="text"><p> 也许在每个恋人眼里对方都是美的，但这个美能维持多久，一小时、一天、一年、还是一辈子！在我的理想中的女人朴实和善良是最美的，美不一定就是外表，想想我们的容颜能维持多久，人的一生很短暂你美丽的容颜只占了生命的五分之一，现实生活中掺杂了太多的无奈，化妆品，手术整形，药物刺激还有无法想象的更多的方式去争取得到那个美，但却失去了纯真的美！
</p></div>
    <div class="text"><p>在我看到一个女孩愿意嫁给曾经救过他人的英雄，但已经面目全非他时，我为此感动，她是最美的！一个女孩因为自己面部在工作受伤而主动提出跟男朋友分手，女孩流着泪说：我不想让你跟我一起承担我的伤痛：男孩简单朴实地说道：你永远是我眼中最美的，就算你到了老的脸上全是皱纹，你也是，嫁给我吧！也许美真的不用刻意去伪装和装饰！</p></div>
    <div class="text"><p>在此我想到了现实生活中，我的身边或多或少地也发生过这样或那样的美，谁能在意到自己身边的过客，我每天看着不同的面孔，想着他们的美丽，清晨忙碌的清洁工你们把街道打扫得如此的美，建筑工人把城市建设得如此美丽，农田里辛勤的农民耕耘出美丽的高粱，而我们排斥或甚至会瞧不起这些美丽，这些美我们都不曾想起。
  </p></div>
</div>
```

(2) 定义 CSS 样式规则。

① 定义页面主体 body 的样式：背景颜色等。样式代码如下所示：

```
     background-color: bisque;
```

② 定义 div 样式：距顶 0px、水平居中、宽度 600px、首行缩进 25px：

```
#div_1{
     width: 600px;
     margin: 0 auto;
     }
```

(3) 定义控制 div 样式：宽度 190px、右边距 10px，向左浮动：

```
.text{
     width: 190px;
     padding: 0 10px;
     float: left;
}
```

(4) 定义段落样式：行距 18px、字号为 14px、字体为“宋体”：

```
p{
    line-height: 18px;
    font-size: 12px;
    font-family: "宋体"
  }
```

4.2.4　运行结果

将写好的页面在浏览器中运行，我们可以看到如图 4-10 所示的效果。



图 4-10　文字编排

4.2.5　实例分析

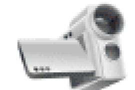


源码解析：

该实例中创建了一个静态页面，在页面中使用 div 布局，简单地使用 div 控制页面的输出版式，然后通过这节所学的几种属性对文本字体进行修饰。最终达到如图 4-10 所示的效果。

4.3　网页文章——感怀秋天

在做网页时，文章是最能突出信息传播的部分，在写文章的基础上，我们使用 CSS 的修饰属性对字体进行修饰，这样使文章读起来有美的享受，看起来也会赏心悦目。

**视频教学：**光盘/视频/4/4.3　网页文章——感怀秋天.avi　　　　长度：6 分钟

4.3.1　基础知识——字体装饰

在本节中我们将讲到对字体的外部修饰所使用的各个属性，包括 text-decoration、

text-underline-position、text-shadow 和 text-transform 属性。这些属性可以对文字进行加下划线、更改字体大小写、添加阴影等操作。下面对这些属性进行介绍。

1. text-decoration

text-decoration 属性可以为页面提供多种文本的修饰效果，比如下划线、闪烁、删除线、贯穿线、上划线等。它的属性值如表 4-4 所示。

表 4-4 text-decoration 属性的取值与描述

属 性 值	效果描述
none	默认值，对文本不进行任何修饰
blink	对文本添加闪烁
underline	对文本添加下划线
overline	对文本添加上划线
line-through	对文本添加贯穿线

语法格式：

```
P{text-decoration: value}
```

或者使用内部修饰语法：

```
<p style=" text-decoration: value ">
```

例如我们对文字添加闪烁和下划线的例子，代码如下：

```
<p  style=" text-decoration: blink ">开夜的天窗，对着划过的流星许下永恒的祝愿，不变的诺言。在每个人的心底，都有最美好的祝愿。或许，这就是一种缘份</p>
 <p  style=" text-decoration: underline ">让我们用最虔诚的心，在这里许下您的心愿、祝福，留下您最想对他(她)说的话，永久地挂在许愿墙上，愿望就会很快实现的呦。</p>
```

将带有代码的文件在浏览器中运行，我们可以看到如图 4-11 所示的效果。

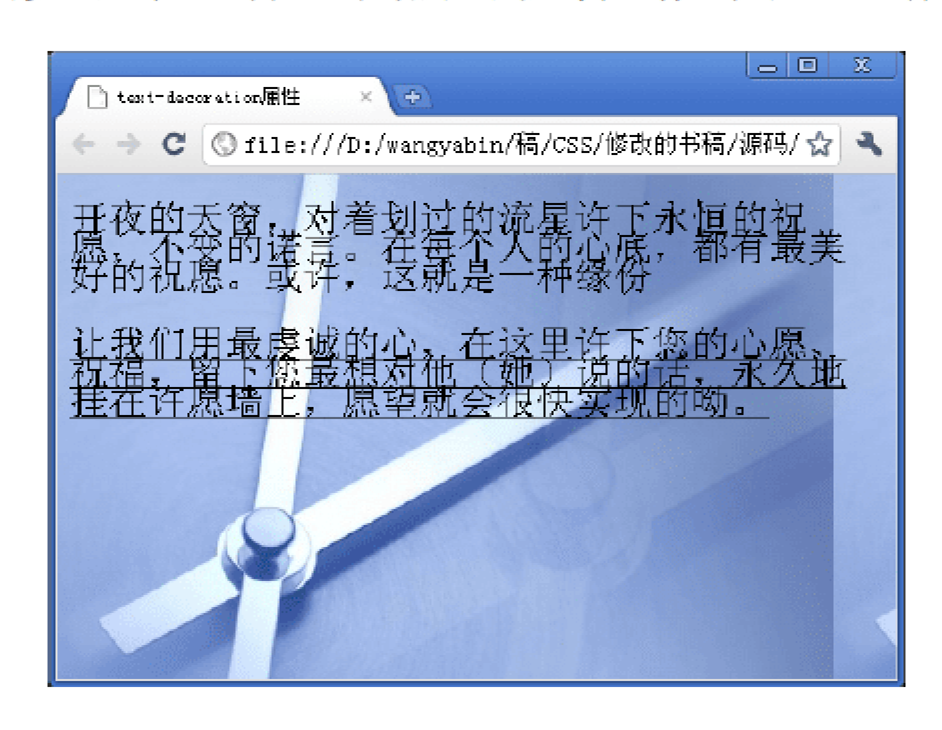


图 4-11 设置 text-decoration 属性的实例运行效果



blink 闪烁效果只有 Mozilla 和 Netscape 浏览器支持，而 IE 和其他浏览器(如 Opera)都不支持该效果。

2. text-underline-position

text-underline-position 属性定义了下划线的位置，此属性对于 currentStyle 对象来说是只读的，对于其他对象而言是可读可写的。它有两个属性值，如表 4-5 所示。

表 4-5　text-underline-position 属性的取值与描述

属 性 值	效果描述
below	默认值，在文本的下面设置下划线
above	在文本上面设置上划线

语法格式：

```
P{text-underline-position: value}或者
<p style=" text-underline-position: value ">
```

例如我们给下面的文字加上下划线和上划线，代码如下：

```
<html>
<head>
<title>text-decoration 属性</title>
<STYLE type="text/css">
p { text-underline-position: above; text-decoration: underline overline; }
</STYLE>
</head>
<body>
    <p > 开夜的天窗，对着划过的流星许下永恒的祝愿，不变的诺言。在每个人的心底，都有最美好的祝愿。或许，这就是一种缘份。<br />
<br />
                让我们用最虔诚的心，在这里许下您的心愿、祝福，留下您最想对他(她)说的话，永久地挂在许愿墙上，愿望就会很快实现的呦。<br />
                </p>
</body>
</html>
```

在浏览器中运行上述文件，会看到如图 4-12 所示的效果。

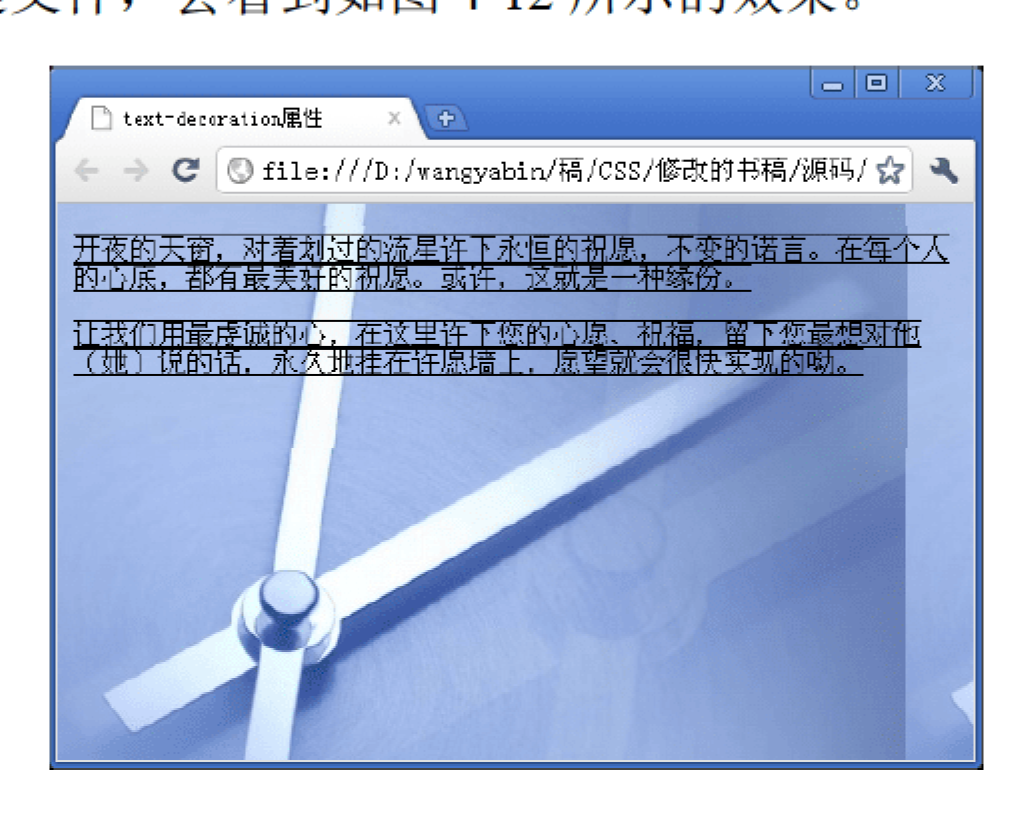


图 4-12　设置 text-underline-position 属性的实例运行效果

3. text-shadow

text-shadow 属性用于设置或检索对象中文本的文字是否有阴影及模糊效果；可以设定多组效果，方式是用逗号隔开；可以用于伪类:first-letter 和:first-line。它的属性值有三个，分别是color、length 和 opacity，效果描述如表 4-6 所示。

表 4-6　text-shadow 属性的取值与描述

属 性 值	效果描述
color	用于指定字体阴影的颜色
length	指定字体阴影的水平和垂直延伸距离，由浮点数和单位标识符组成的长度值，可为负值
opacity	指定模糊效果的作用距离，如果仅仅需要模糊效果，将 length 设为 0 即可，由浮点数和单位标识符组成的长度值，不可为负值

语法如下：

```
text-shadow : none | <length> none | [<shadow>, ] * <shadow> 或 none | <color>
[, <color> ]*
```

下面我们来看看这三种属性值的效果吧。代码如下：

```
<title>text-shadow 属性</title>
<STYLE type="text/css">
H2 { text-shadow: 3px 3px red, yellow -3px 3px 2px, 3px -3px }
</STYLE>
</head>
<H1>
    Dreams
  </H1>
  <div style="text-shadow:5px 2px 6px #000;">
    <p >Hold fast to dreams For if dreams die Life is a broken-winged bird That
cannot fly </p>
    <p >Hold fast to dreams For when dreams go Life is a barren field Frozen
with snow </p>
  </div
```

在 IE 6、IE 7、IE 8 版本中都不支持 text-shadow 属性，在 Chrome 浏览器下运行的效果如图 4-13 所示。



图 4-13　设置 text-shadow 属性的实例运行效果

4. text-transform

text-transform 属性设定文本字体的大小写转换。该属性有 4 种属性值，如表 4-7 所示。

表 4-7　text-transform 属性的取值与描述

属 性 值	效果描述
none	默认值，对文本无任何转换
capitalize	将每个单词的第一个字母转换成大写，其余无转换发生
uppercase	转换成大写
lowercase	转换成小写

该属性仅作用于字符型文本，语法如下：

```
text-transform : none | capitalize | uppercase | lowercase
```

下面通过一个实例来演示大小写转换的效果，代码如下：

```
<html>
<head>
<style type="text/css">
  h1 {text-transform: none}
  p.uppercase {text-transform: uppercase}
  p.lowercase { text-transform: lowercase}
  p.capitalize {text-transform: capitalize}
</style>
</head>
<body>
<h2>
<p class="uppercase">Dreams</p>
<p class="lowercase">Hold fast to dreams For if dreams die Life is a broken-winged
bird That cannot fly </p>
<p class="capitalize">Hold fast to dreams For when dreams go Life is a barren
field Frozen with snow </p></h2> </body>
</html>
```

在浏览器中查看该页面的效果，如图 4-14 所示。

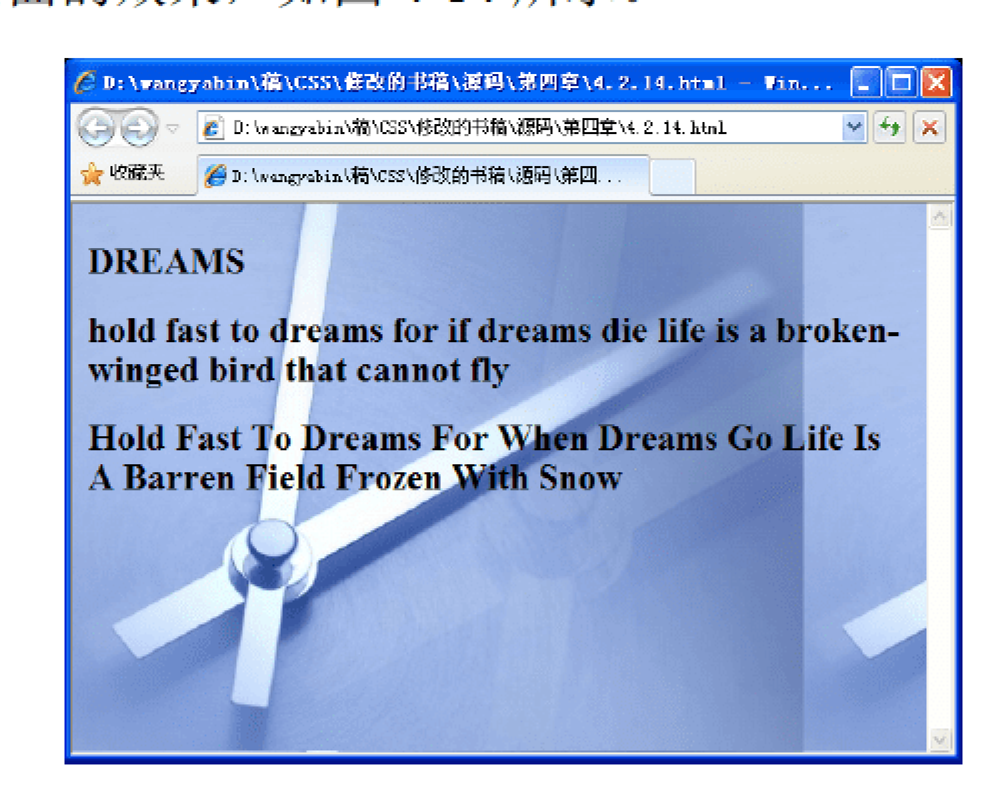


图 4-14　设置 text-transform 属性的实例运行效果

4.3.2 实例描述

在设计个性网站时，常常希望能够突出地显示主要信息，比如使用字体的装饰效果、阴影等，下面就来实现一个有个性的网页文章吧。

4.3.3 实例应用

【例 4-3】网页文章——感怀秋天

在这个实例中，我们通过字体装饰属性 text-decoration、text-shadow、text-underline-position、text-transform 来对网页中的字体进行修饰，以达到很好的视觉效果。下面我们将一步一步地实现这个实例。

(1) div 布局页面。为了达到效果我们使用 div 进行布局，把一个 div 放入#div_body 里，使用#div_left 来控制属性，代码如下：

```
<div id="div_body">
  <p> </p>
  <div id="div_left">
  <h1 style="text-shadow:5px 2px 6px #000;">感怀秋天</h1>
  <h3>季节变换，生命轮回，风瑟雨萧，秋色覆没了春的蓬勃茁壮和夏的缤纷绚丽，又一个秋天来了。天高高的，云闲闲的，百花凋零众芳飘落，生命在不经意间慢慢枯去，一切都在悄悄地改变。自古逢秋悲寂寥，秋天是一个淡泊的时节，是一个善于感怀的时节。
  </h3>
  </div>
</div>
```

(2) 定义 CSS 样式规则。

① 定义主体 body 样式。设置背景颜色为#FAF3D8，代码如下：

```
body {
    background-color: #FAF3D8;
}
```

② 定义 div 样式。如添加背景图片、div 的位置等。以下是页面中两个 div 的样式代码：

```
#div_body {
    width: 670px;
    height:600px;
    background-image:
url(../../%E8%83%8C%E6%99%AF%E5%9B%BE%E7%89%87/http_imgload24.jpg);
    background-attachment: scroll;
    margin-left:170px;
}
#div_left{
    margin-top: 150px;
    width: 400px;
    height: 400px;
     }
```

③ 定义文章标题样式。如字体类型、大小等，代码如下：

```
<h1 style="text-shadow:5px 2px 6px #000;">感怀秋天</h1>
h1{
    font-family: "宋体";
    font-size: 24px;
    font-style: normal;
    font-weight: bolder;
    text-transform: none;
    text-decoration: underline;
    color: #F60;
    text-align: center;
    }
```

④ 定义文章主体样式。如字体大小、字体阴影等，代码如下：

```
h3{
    font-family: "新宋体";
    font-size: 18px;
    font-style: italic;
    line-height: 60px;
    font-weight: 200;
    font-variant: normal;
    text-transform: none;
    color: #F33;
    text-shadow:5px 2px 6px #000;
    }
```

4.3.4　运行结果

将写好的页面在浏览器中运行，效果如图 4-15 所示。

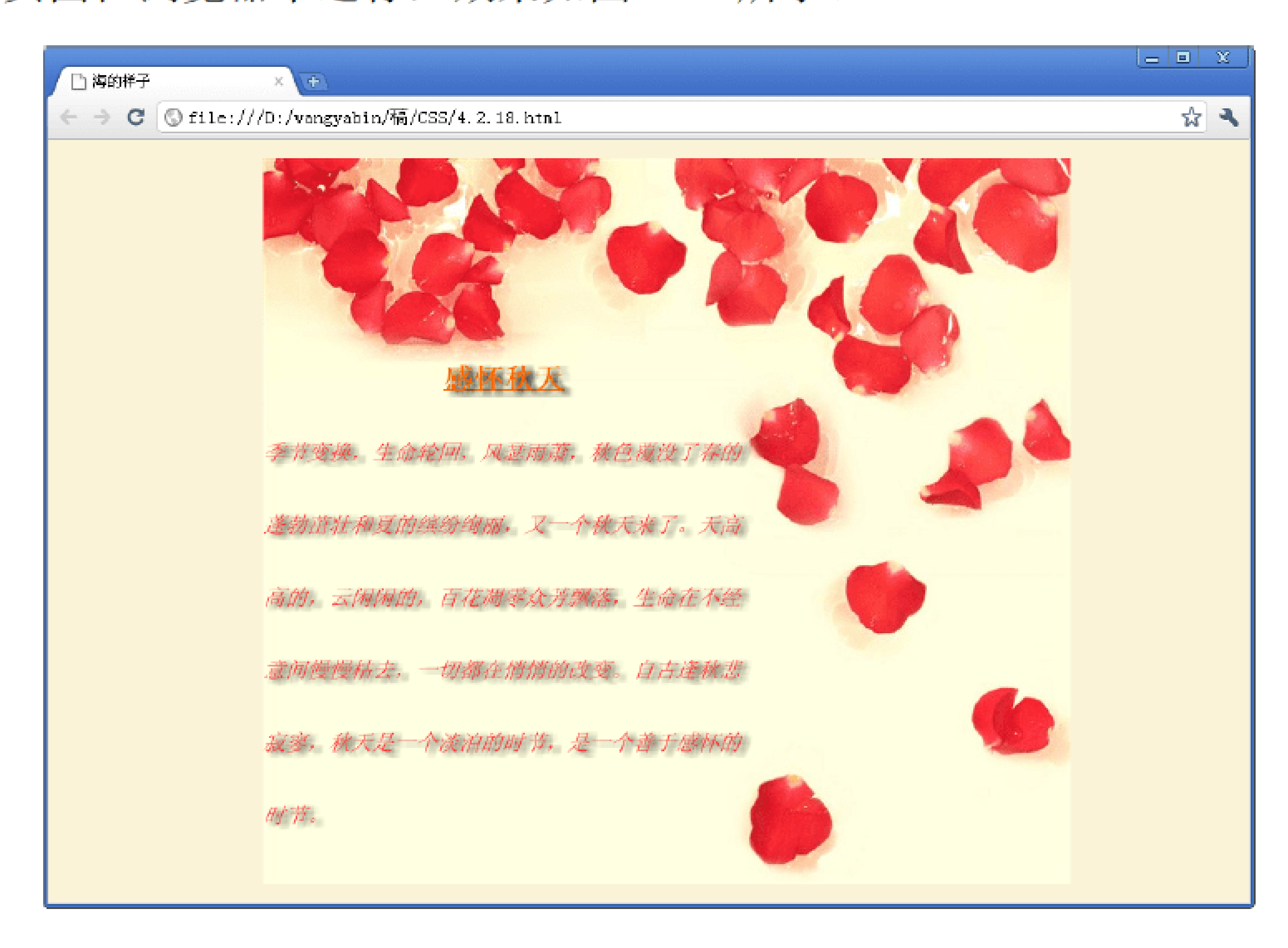


图 4-15　网页文章实例运行效果

4.3.5 实例分析

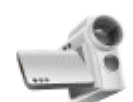


源码解析：

在本实例中，采用了 div 布局和 CSS 字体的阴影、下划线等效果对文字进行修饰，才达到我们需要的效果。字体的这些属性可以使我们的文章在一个页面中以很好的结构和独特样式展现出来。

4.4 设置字体间隔

在设计网页时，字体的间隔设置为多大才能使页面更加美观，更出效果。这是网页设计人员需要重视的一个问题。



视频教学：光盘/视频/4/4.4 Welcome to Beijing.avi 长度：3 分钟

4.4.1 基础知识——字体间隔

在这里我们需要用到字体的间隔属性 letter-spacing 和 word-spacing，这两个属性都适用于页面所有的元素。

1. letter-spacing

letter-spacing 属性用于设定字符间距，将指定的间隔添加到每个文字之后，但最后一个字将被排除在外。它有两个属性值 normal 和 length，如表 4-8 所示。

表 4-8 letter-spacing 属性的取值与描述

属 性 值	效果描述
normal	默认值，设置默认间隔
length	由浮点数字和单位标识符组成的长度值，允许为负值

使用 letter-spacing 属性可以对字符进行缩放，如果在该属性中使用负值，则字符将重叠在一起，如以下代码：

```
<body>
<h2>
<p style="letter-spacing:normal">正常字符间距</p>
<p style="letter-spacing:8px">字符间距为 8</p>
<p style="letter-spacing:3px">字符间距为-3，字符就重叠在一起</p>
</h2>
</body>
```

在浏览器中运行这些代码的效果如图 4-16 所示。

2. word-spacing

word-spacing 属性用来设置词与词之间的间距，它也有 normal 和 length 两个属性值，如表 4-9 所示。



图 4-16　设置 letter-spacing 属性的实例运行效果

表 4-9　word-spacing 属性值

属 性 值	效果描述
normal	默认值，设置默认间隔
length	由浮点数字和单位标识符组成的长度值，允许为负值

word-spacing 属性可以接收正的或负的长度值，该长度值用于设定文本中词与词之间的距离。该属性的默认值为 normal，表示间距为 0。例如，有如下样式规则定义单词之间的间距，代码如下：

```
<p style="word-spacing:normal">Welcome to my CSS</p>
<p style="word-spacing:10px">Welcome to my CSS</p>
```

在浏览器中查看页面效果如图 4-17 所示。

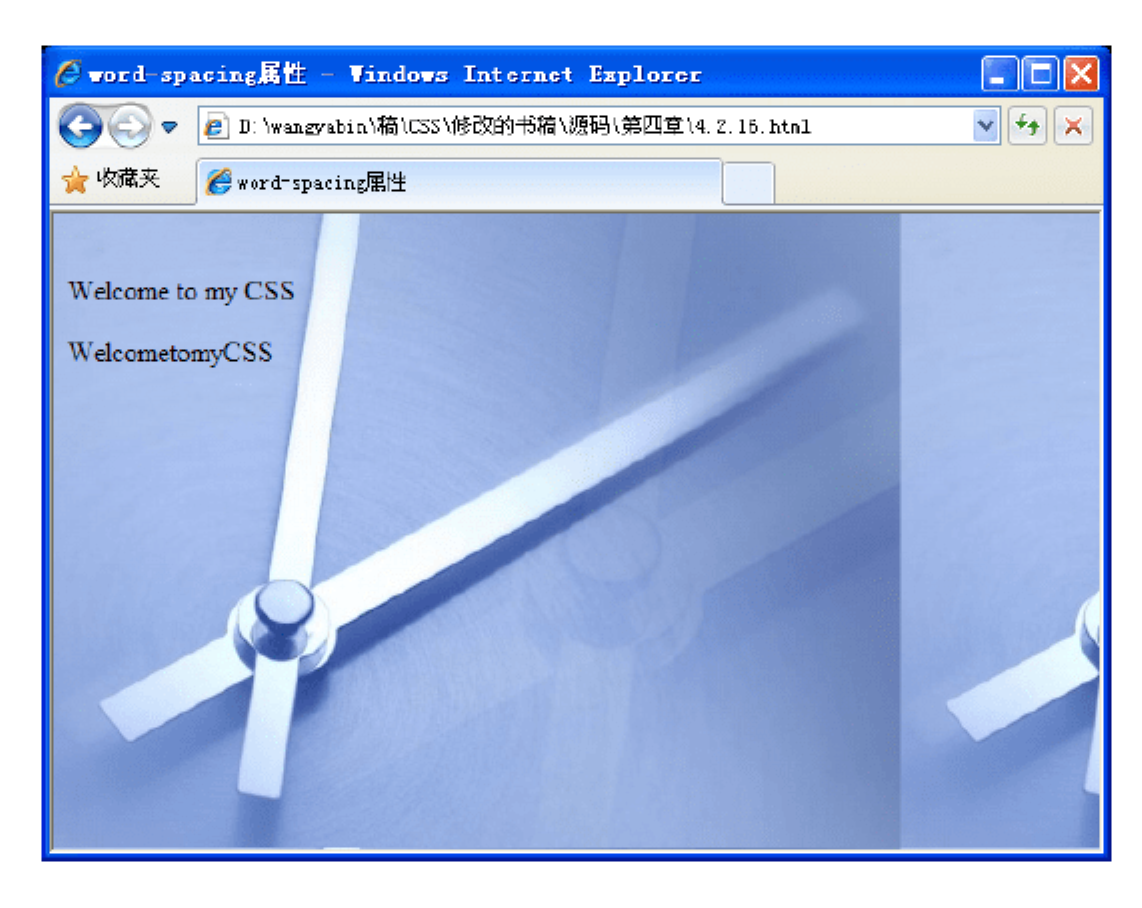


图 4-17　设置 word-spacing 属性的实例运行效果

如果在该属性值中使用负值，将会使单词之间更加紧凑，如设置 word-spacing 属性值为−8px，效果如图 4-18 所示。

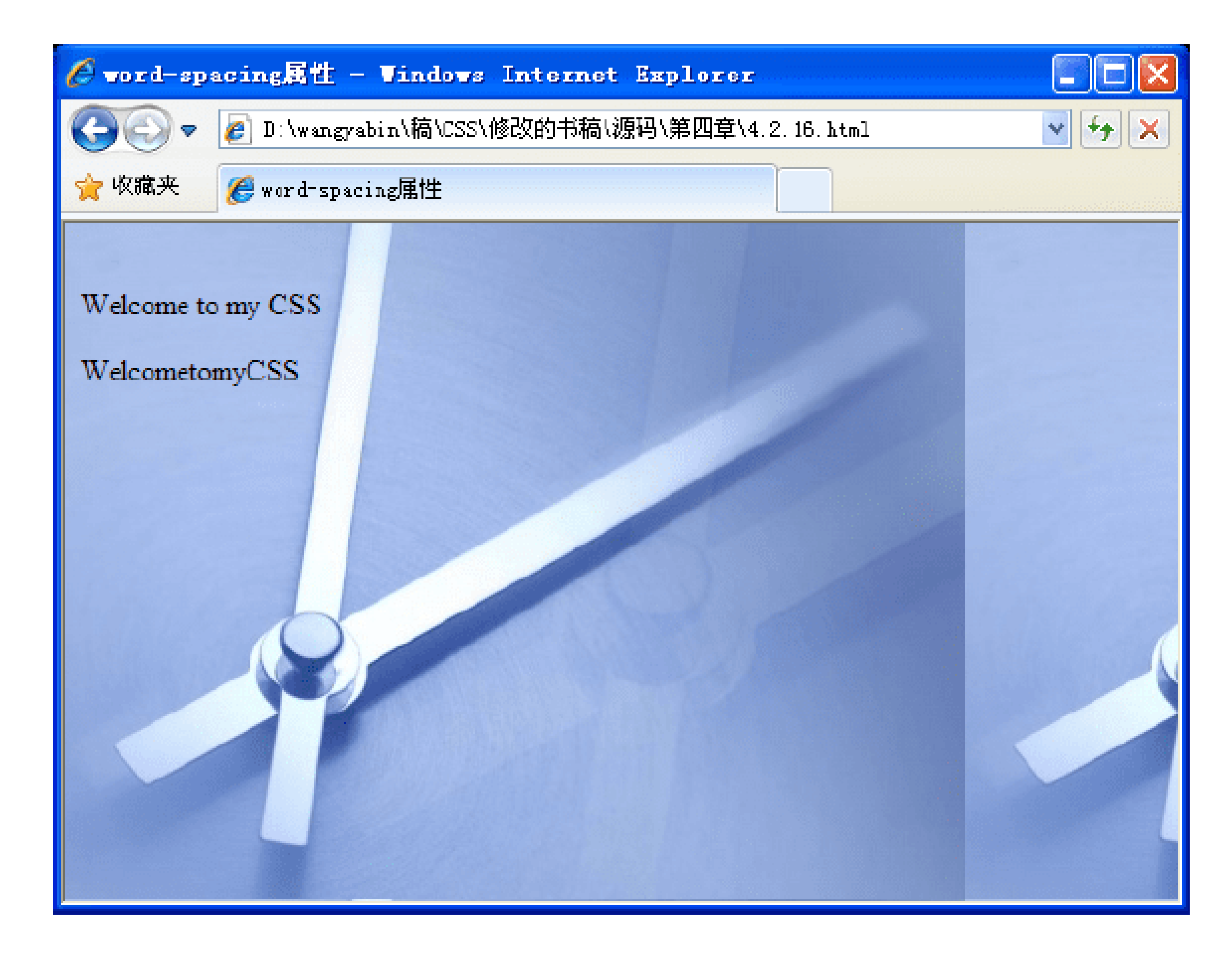


图 4-18　单词间距为负值的实例运行效果

4.4.2　实例描述

字体间隔属性在页面中时常会用到，可以根据人的阅读习惯，适当地调整字体的间隔。下面我们来看一个实例，通过这种实例，主要学习 CSS 的字体间隔属性。

4.4.3　实例应用

【例 4-4】Welcome to Beijing

在这个实例中，通过 word-spacing、letter-spacing 属性来控制文章中字体与字体，词与词之间的间隔，下面将实现这个实例。

(1) 文章标题样式：定义字符间距为 5px，内容放在<h1>标签内，代码如下：

```
h1{
    letter-spacing:5px;
}
```

(2) 定义文章内容样式：设置单词间距为 3px，代码如下：

```
p{
    word-spacing:3px;}
```

4.4.4　运行结果

将编写好的页面在浏览器中运行，结果如图 4-19 所示。

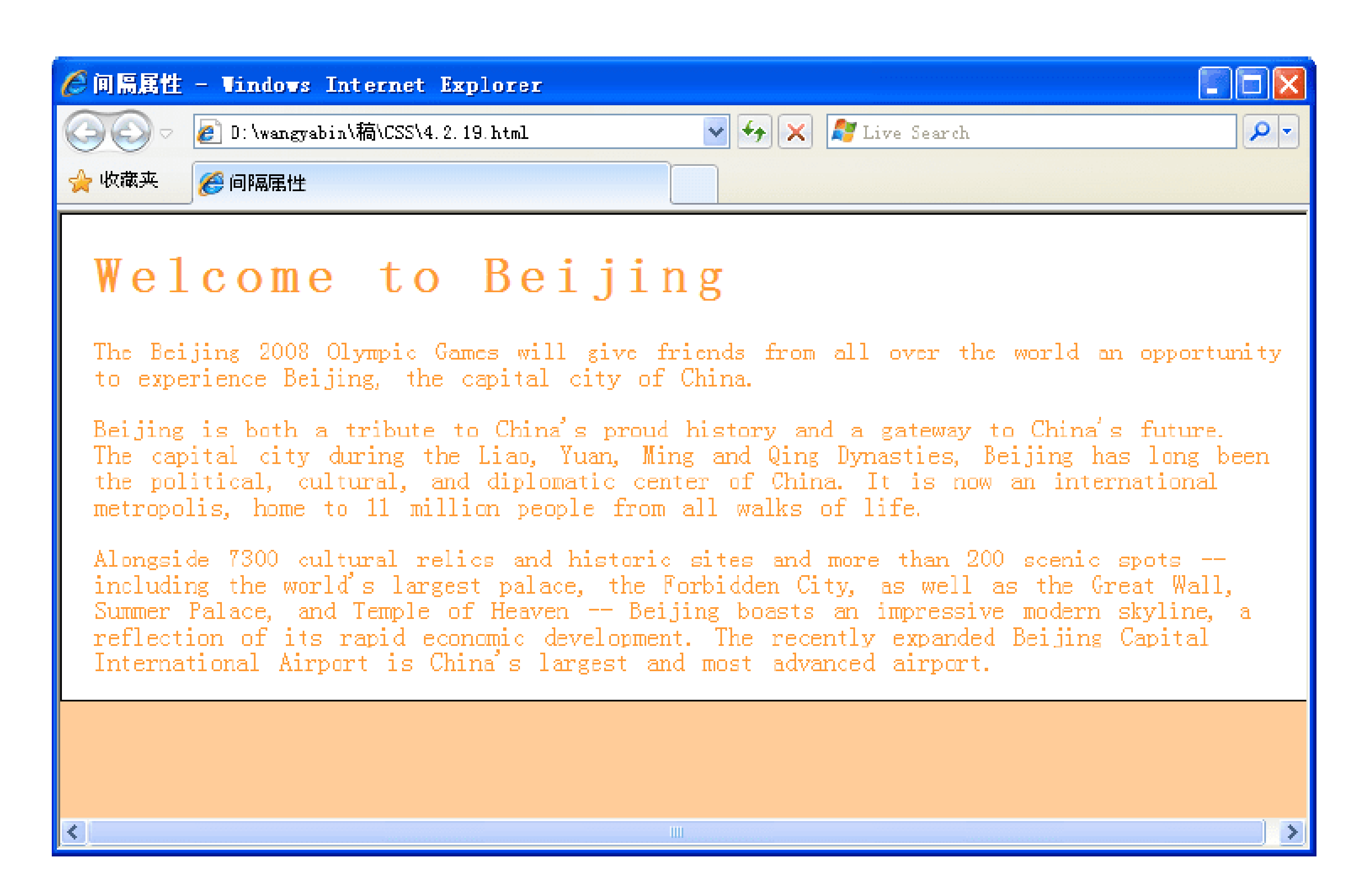


图 4-19　间隔实例运行效果

4.4.5　实例分析



源码解析：

在本实例中，主要讲解字体间隔，页面的布局和上一节相同，使用 div 进行布局和控制。在页面中，文章标题部分使用 letter-spacing 来定义字符间隔样式。

4.5　行　　高

大片密密麻麻的文字往往会让人觉得乏味，因此适当地调整行高(line-height)可以降低阅读的困难与枯燥，并且使页面显得美观。



视频教学：光盘/视频/4/4.5　行高.avi　　**长度：**5 分钟

4.5.1　基础知识——字体行高

行高指的是文本行的基线间的距离，但是文本之间的空白距离不仅仅是由行高决定的，同时也受字号的影响。我们可以通过 CSS 的 line-height 属性来设置行高。

line-height 属性用来设置行间距。事实上，这一属性决定了内容行垂直高度所能增减的数值。那么，行间距到底是如何应用的呢？

网页中的每个元素都有一个内容区域，该内容区域是由字体尺寸决定的。内容区域中又包含了内联框，内联框的范围就是内容区域。行间距则可以增减内联框的高度，这是通过将行间

距分为两半，再将每个半行间距应用于内容区域的顶部和底部来实现的，而这又构成了内联框。例如，设定内容区域为 20px，行间距为 30px，则两者的差值 10px 将分为两半，将每一半(5px)分别应用到内容区域的顶部和底部，就得到 30px 高的内联框。简而言之，行间距就是内容区域最底端与其内部顶端之间的距离。

line-height 的属性值也有两个，分别是 normal 和 length，如表 4-10 所示。

表 4-10　line-height 属性的取值与描述

属 性 值	效果描述
normal	默认值，设置默认行高
length	百分比数值和由浮点数字和单位标识符组成的长度值，允许为负值，其百分比取值是基于字体的高度尺寸

在上面的属性值中，最常用的是直接设置高度和使用默认值，代码如下：

```
<div class="box_grey" style="font-size:18px">
                <h3><img src="images/page_notice.gif" width="19px"
height="16px" alt="textbox" /> <strong>神奇亚运会</strong></h3>
            <p style="line-height:normal">亚运，曾经在多少人心中燃烧过希望，在多少人的手中传递过梦想。因为这个梦想，有火热的情感，有奔涌的热血，有苦甜的汗水。每一个路标都刻下关于梦想的预言；因为这个梦想，用微笑迎接闪电，用坦然面对狂风，心底里不熄的永远是那一炬炽热的火炬，跳动的火焰。
</p>
<p style="line-height:50px">亚运，曾经在多少人心中燃烧过希望，在多少人的手中传递过梦想。因为这个梦想，有火热的情感，有奔涌的热血，有苦甜的汗水。每一个路标都刻下关于梦想的预言；因为这个梦想，用微笑迎接闪电，用坦然面对狂风，心底里不熄的永远是那一炬炽热的火炬，跳动的火焰。
</p>
```

设定行间距的单位有 px、pt、cm、mm 等，用户可以使用不同的单位进行设置。在浏览器中查看页面，如图 4-20 所示。

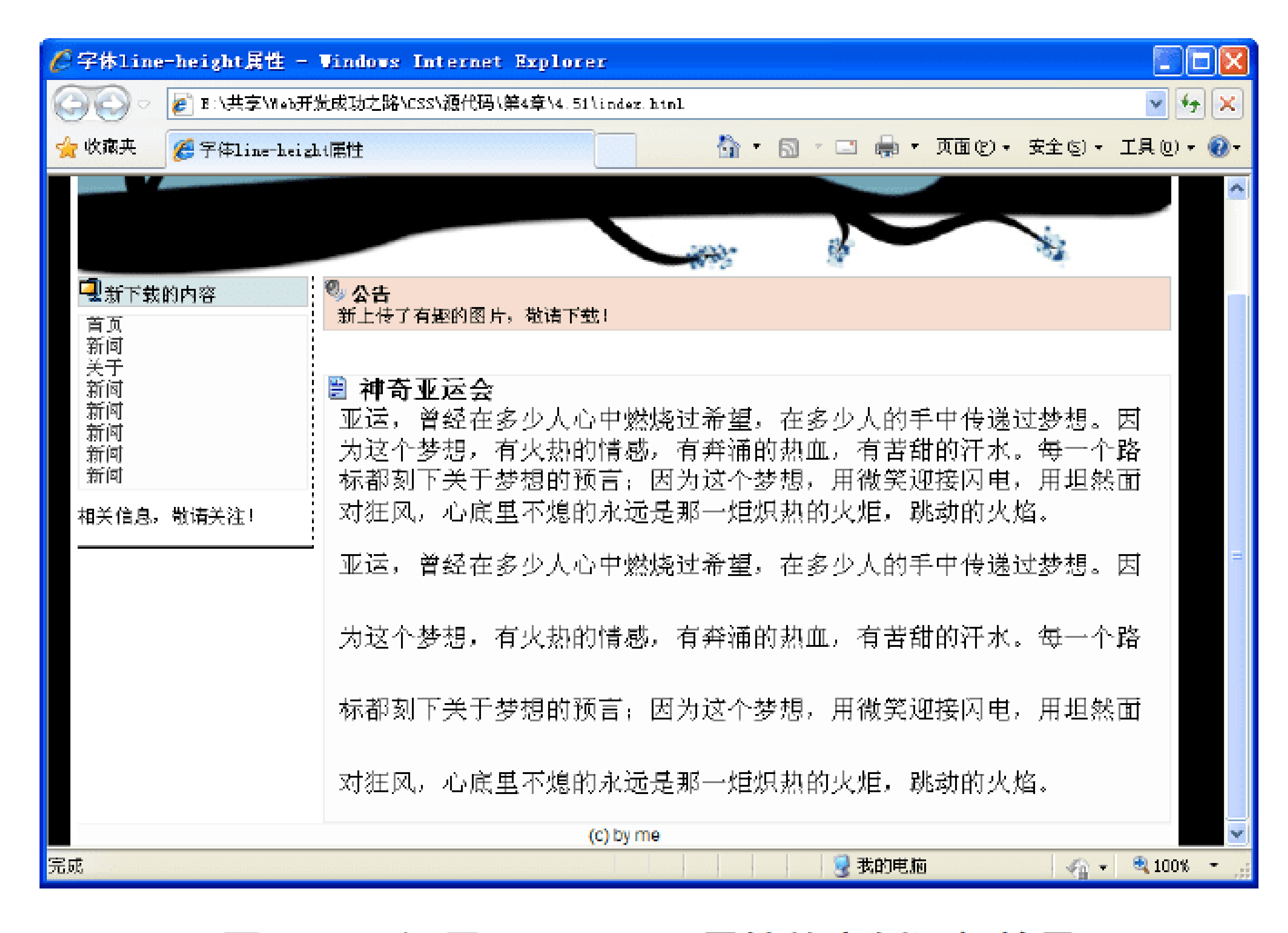


图 4-20　设置 line-height 属性的实例运行效果

4.5.2　实例描述

我们在看新闻网页时，新闻版面段落的行与行之间的间距非常恰当，这一般是通过 CSS 行高属性来控制的，下面就简单地做一个新闻版面。

4.5.3　实例应用

【例 4-5】新闻版面

我们所要学习的版面的控制格式还是通过 div 来控制的，这个实例主要通过对行高的控制，来使文章段落看起来美观，下面来一步步地实现吧。

(1) div 布局页面。在 div1 里放入上下两层 div，分别是 div_top 和 div_xia，下层 div 里又分两层 div，分别为 div1 和 div2，这两层写入新闻的内容。代码如下：

```
<div id="div1">
  <div id="div_top"><table width="100%" border="1">
  <tr>
    <td align="center"><h1 id="artibodyTitle" pid="1" tid="1" did="21602876"
fid="1666">阿桑奇被捕后维基解密再曝光近百份文件</h1></td>
  </tr>
</table>
</div>
<div id="div_xia">
<div id="div_1">
<p>环球网记者梁旭报道 12 月 7 日，维基解密创始人阿桑奇在伦敦遭拘捕后该网站再次曝光最新文件，包括 97 份美国议会的最新机密电文。俄塔斯社报道称，上述文件涉及美国驻马德里大使馆和驻里约热内卢领事馆的相关工作情况。在阿桑奇被捕后，维基解密网站表示仍将继续公布他们掌握的 251287 份涉及美国议会的文件，但这
需要几个月时间。</p>
  <p>  俄塔斯社报道称，上述文件涉及美国驻马德里大使馆和驻里约热内卢领事馆的相关工作情况。在阿桑奇被捕后，维基解密网站表示仍将继续公布他们掌握的 251287 份涉及美国议会的文件，但这需要几个月时间。</p>
<p>
报道称，维基解密曝光的美国外交官电报文件中 15652 份属于“机密”，101748 份文件属于“保密”信息。</p>
</div>
<div id="div_2">版权所有某某公司
</div>
</div>
</div>
```

(2) 定义 div 的样式。主要定义字体大小、颜色、行高及布局等样式。代码如下：

```
#div1 {
    font-size: 18px;
    font-style: normal;
```

```
    line-height: 10px;
    font-weight: bold;

}
#div_top{
    font-family: "宋体";
    font-size: 18px;
    line-height: normal;
    font-weight: 700;
    font-variant: normal;
    text-transform: none;
    color: #FFF;
}
#div_xia{
    font-family: "新宋体";
    font-size: 18px;
    line-height:30px;
    font-weight: bold;
    color: #CCC;
    position: absolute;
    text-align: center;
}
#div_1{
    text-align: center;
}
```

基本的样式就这么多，可以看到在#div_xia 中我们定义行高属性 line-height 为 30px。

4.5.4　运行结果

将做好的网页在浏览器中运行，可以看到以下效果，如图 4-21 所示。

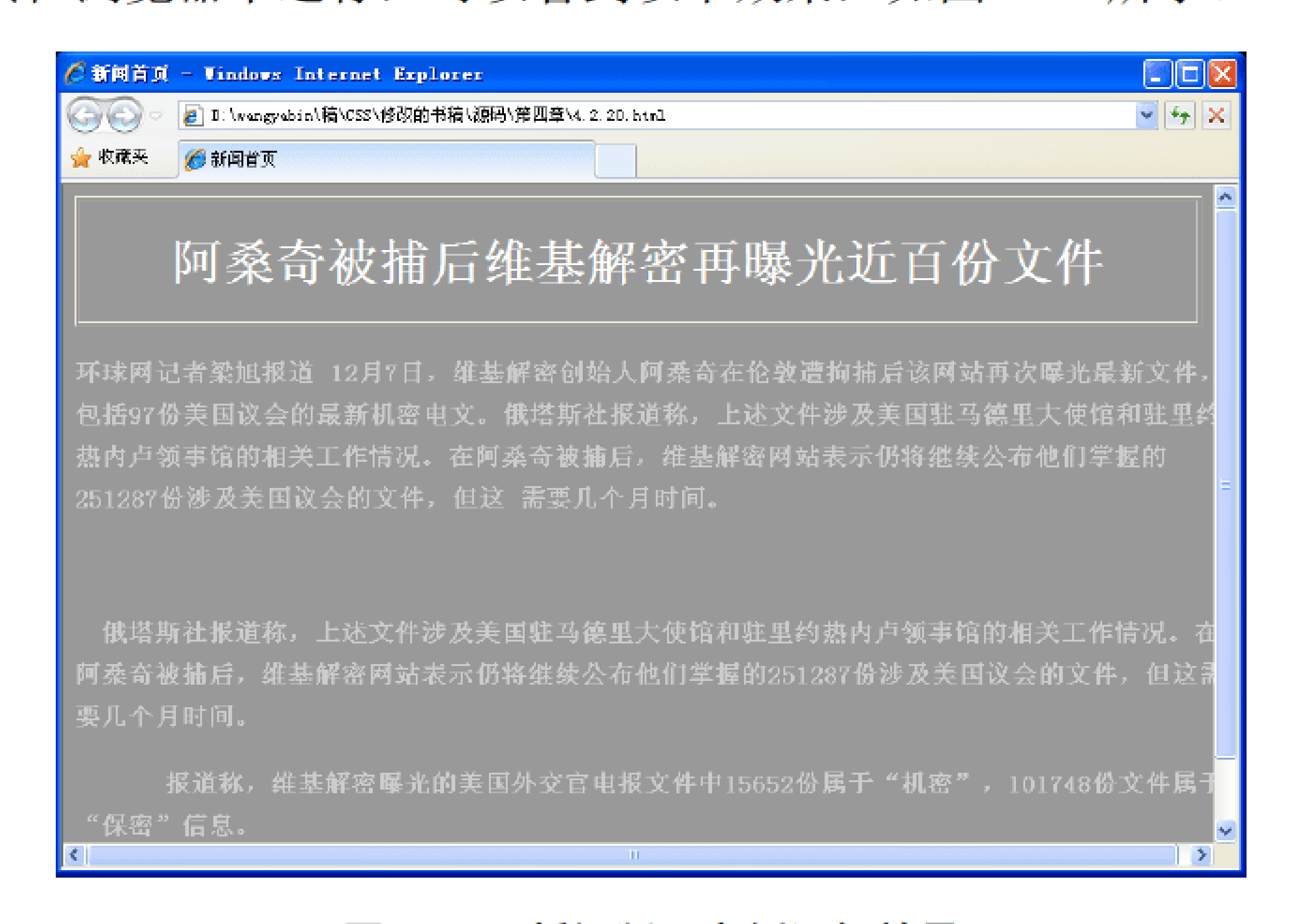


图 4-21　新闻版面实例运行效果

4.5.5　实例分析



源码解析：

以上实例我们通过 div 和 CSS 的样式实现一个新闻版面，在此页面中我们单独对行元素添加样式，也就是 CSS 的 line-height 属性。通过此属性可以控制行与行之间的高度。

4.6　常见问题解答

4.6.1　字体名称样式的问题



字体名称样式 font-family 的问题。
网络课堂：http://bbs.itzcn.com/thread-12407-1-1.html

网站导航我想采用华文彩云或幼圆字体或其他字体，怎么让它表现得和 Photoshop 里的形式一样呢(边缘也很光滑的那种)？我现在设置的字体加粗了但边缘都很粗糙，不知为什么？

【解决办法】首先肯定地告诉你，绝对不能用 CSS 来制作任何特殊的非系统字体，你自己装的那些别人是看不见的。我们最好使用 CSS 中的滤镜来解决。

4.6.2　italic 和 oblique 的问题



italic 和 oblique 的问题。
网络课堂：http://bbs.itzcn.com/thread-12418-1-1.html

italic 和 oblique 的英文意思一样，网页中显示一样，为什么弄俩一样的？请高手讲讲，谢谢！

【解决办法】italic 是斜体文本，它除了倾斜 normal 文本外，可能会在每个字母上做点小变化以获得外观的变化；oblique 是倾斜文本，它只是 normal 文本的倾斜。有些字体有这两种字体的区别，但是大部分字体不区分 italic 和 oblique，即使有字体上的区分，也很少有浏览器能够区分它们之间的差别。

4.6.3　字体大小问题



字体大小写代码问题。
网络课堂：http://bbs.itzcn.com/thread-12419-1-1.html

以下是我写好的 CSS 代码：

```
body{
   background:#F6FEE9;
 }
#container {
 background:url(img/top.jpg) repeat fixed center 0;
 }
```

该代码可以实现在 IE9 中显示一张背景图片，但是现在如何加入些代码使我可以控制网页字体？在哪个地方加？

【解决办法】根据文字所在的位置来确定 CSS 中的选择器。例如有一段文字“<p>你好</p>”，那么要想在 CSS 里控制字体就应该添加这样的 CSS 代码“p{font-size:属性值；font-family:属性值；font-color:属性值；}”。这行 CSS 代码可以控制字号、字体、颜色的。当然其它的文本属性可以模仿着写上就可以了。

4.7 习　　题

1. 填空题

(1) font-style 属性用于定义字体显示的样式，它的三种可选属性值是 normal、italic、__________。

(2) 在设置字符间距时应该用__________属性。

(3) __________是设置字体的综合属性。

2. 选择题

(1) 以下关于 font 标记符的说法中，错误的是__________。

A. 可以使用 color 属性指定文字颜色

B. 可使用 size 属性指定文字大小

C. 指定字号时可以使用 1～7 的数字

D. 语句“这里是 2 号字”将使文字以 2 号字显示

(2) 用__________实现字体的阴影及模糊效果。

A. text-decoration　　B. font-weight

C. text-shadow　　D. font-family

(3) 以下__________不是 text-transform 的属性值。

A. lowercase　　B. opacity

C. uppercase　　D. capitalize

3. 上机练习

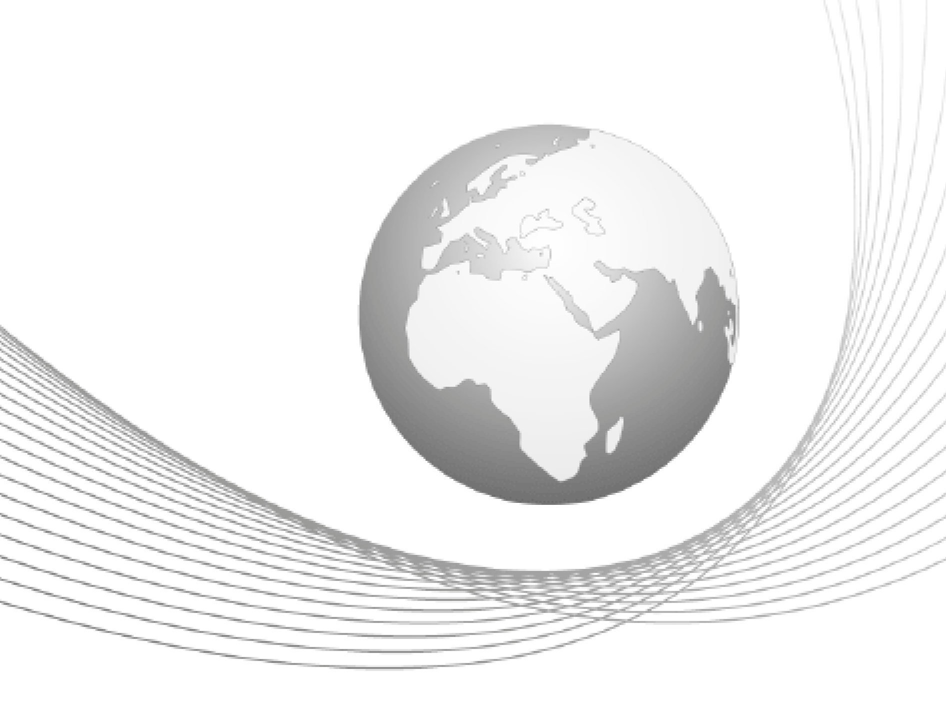
上机练习：使用 Dreamweaver 创建一个简单的页面，其中要用到字体的属性。

通过本章的学习，我们知道字体在页面中的应用是最多的，本次上机练习也主要是针对字体属性的。

要求读者实现如图 4-22 所示的页面效果。



图 4-22　练习效果



第 5 章　文 本 属 性

内容摘要：

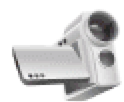
文本表示的是页面所包含的内容，文本的放置与效果的显示会直接影响到页面的布局以及风格。CSS 样式提供了文本属性用于实现对页面中文本的控制显示。本章将对几种重要的文本属性进行介绍。

学习目标：

- 掌握文本对齐方式属性的应用
- 掌握文本内容流的方向应用
- 掌握文本的排序应用
- 掌握文本的其他样式

5.1 推荐图书详细信息

决定段落边缘的外观和方向：左对齐、右对齐、居中、两端对齐和分散对齐。例如，在一个左对齐段落中(最常见的对齐方式)，段落的左边缘是和左页边距对齐的，本节将对 text-indent 属性、vertical-align 属性、text-align 属性、white-space 属性和 ruby-align 属性进行讲解，同时借助推荐图书详细信息实例进行讲解。

**视频教学：**光盘/视频/5/5.1 推荐图书详细信息 长度：6 分钟

5.1.1 基础知识——text-indent 属性

在网页设计中，段落首行缩进是必不可少的样式。因此 CSS 也考虑到了这点，通过使用 text-indent 属性可以实现文本首行缩进的效果。

text-indent 属性中有两种不同的缩进方式，一种直接定义缩进长度，另一种则是定义缩进的百分比，详细代码如下所示。

```
<p style="text-indent:10mm">XML 是一种描述数据和数据结构的语言，可以保存在任何可以存储文本的文档中。本书主要介绍了 XML 的产生历史和特点，XML 的基本语法和编辑工具，XML 的运行环境，约束性文档 DTD，XML Schema，XML 在显示时的样式语言 CSS 和 XSL，XML 文档接口 DOM、SAX，XLink 和数据岛技术。</p>
<p style="text-indent:10%"> XML 是一种描述数据和数据结构的语言，可以保存在任何可以存储文本的文档中。本书主要介绍了 XML 的产生历史和特点，XML 的基本语法和编辑工具，XML 的运行环境，约束性文档 DTD，XML Schema，XML 在显示时的样式语言 CSS 和 XSL，XML 文档接口 DOM、SAX，XLink 和数据岛技术。</p>
```

text-indent 属性还有一种特殊的缩进方式——悬挂缩进。它可以使首行悬挂在标签的左边，要实现该效果只需要设置属性的值为负数即可，详细代码如下所示。

```
<p style="text-indent:-10mm">XML 是一种描述数据和数据结构的语言，可以保存在任何可以存储文本的文档中。本书主要介绍了 XML 的产生历史和特点，XML 的基本语法和编辑工具，XML 的运行环境，约束性文档 DTD，XML Schema，XML 在显示时的样式语言 CSS 和 XSL，XML 文档接口 DOM、SAX，XLink 和数据岛技术。</p>
```

上述代码中如果属性值为负数，则前面的部分内容被浏览器窗口挡住，我们可以设置 padding-left 属性，使内容显示完全。

在布局中如果上级标签标记定义了 text-indent 属性，那么在该标签之内的所有标签则都会继承上级标签标记定义的属性值。



text-indent 属性属性值可以使用绝对长度单位，例如 cm、mm 等，还可以使用相对长度单位，例如 em、px 等。

5.1.2　基础知识——vertical-align 属性

在 XHTML 中，<sup>和<sub>标签用来显示页面中的上标和下标。同样 CSS 也有这样的属性来控制显示内容的垂直对齐方式，那就是 vertical-align 属性。该属性仅对 table 有效，或者可以说是仅对 td 有效。div、ul、li、p 等容器的竖直方向永远是顶端对齐的。

vertical-align 属性有 9 个预设值，用来控制标签内容的对齐方式，属性值如下所示。

- auto：根据 layout-flow 属性的值对齐对象内容。
- baseline：默认值。将支持 valign 特性的对象的内容与基线对齐。
- sub：垂直对齐文本的下标。
- super：垂直对齐文本的上标。
- top：将支持 valign 特性的对象顶端对齐。
- text-top：将支持 valign 特性的对象的文本与对象顶端对齐。
- middle：将支持 valign 特性的对象的内容与对象中部对齐。
- bottom：将支持 valign 特性的对象的内容与对象底端对齐。
- text-bottom：将支持 valign 特性的对象的文本与对象底端对齐。

为了更好地让属性值在页面上显示理想的效果，这次将通过使用上下标、顶端对齐、底部对齐三种方式进行讲解。

1．上下标方式

首先简单地介绍一下基线对齐方式 baseline。基线对齐方式在页面中同其他文本内容的显示差别不大。上下标在页面的数学运算或注释标号中使用得比较多，详细代码如下所示。

```
<p>
    北京<b style="font-size:pt;vertical-align:super">2008</b>
    奥运<b style="font-size:8pt;vertical-align:sub">[注释]</b>
    加油！<img src="1.gif" style="vertical-align:baseline">
</p>
```

在实际应用中，上下标对齐方式的字体通常情况下都会比正常的字体小，它主要起到一个标注的作用。

2．顶端对齐方式

顶端对齐又分为两种参照方式，一种是参照整个文本块，一种是参照文本。下面我们通过一个简单的代码了解一下该如何使用顶部对齐方式，详细代码如下所示。

```
<p>
    <img src="xin.gif" style="vertical-align:middle"/>
        北京！奥运！加油！<img src="1.gif" style="vertical-align:top">
</p>
<hr/>
<p>
    <img src="xin.gif" style="vertical-align:middle"/>
```

```
    北京！奥运！加油！<img src="1.gif" style="vertical-align:text-top">
</p>
```

在上述代码中，我们采用了 middle 中间对齐方式。将第一个段落中第二张图片的 vertical-align 属性值为 top，表示该图片要与包含它的文本块顶端对齐。将第二个段落中的第二张图片的 vertical-align 值设置为 text-top，表示该图片要与所在的文本块中的文本顶端对齐。

3. 底部对齐方式

底部对齐方式和在上面讲述的顶部对齐方式一样，分别参照文本块与文本块中包含的文本块为实例，详细代码如下所示：

```
<p>
    <img src="xin.gif" style="vertical-align:middle"/>
    北京！奥运！加油！<img src="1.gif" style="vertical-align:bottom">
</p>
<hr/>
<p>
    <img src="xin.gif" style="vertical-align:middle"/>
    北京！奥运！加油！<img src="1.gif" style="vertical-align:text-bottom">
</p>
```



vertical-align 属性不能应用于表格单元格中的内容的对齐。

同时，vertical-align 属性值还能使用百分比来设定垂直的高度，它的高度具有相对性，是基于行高的值来计算的，而百分比还可以使用正负号来表示，正号百分比表示文本上升，负号百分比表示文本下降，详细代码如下所示：

```
<p>
    北京<b style="font-size:pt;vertical-align:100%">2008</b>
    奥运<b style="font-size:8pt;vertical-align:-100%">[注释]</b>
    加油！<img src="1.gif" style="vertical-align:baseline">
<p>
```

5.1.3 基础知识——text-align 属性

text-align 属性的主要作用是用来设置文本的水平对齐方式。同样它也有 4 种对齐方式：left、center、right 和 justify。

- left：表示向左对齐。
- center：文本居中。
- right：表示向右对齐。
- justify：表示两端对齐方式。

当我们为其指定值时，默认对齐方式是向左对齐。

通常情况下，居中对齐方式使用在标题类的文本中，其他对齐方式可以根据页面布局来选

择使用，下面通过一些简单的实例来讲解如何在页面中使用，详细代码如下所示：

```
<body>
    <h1 style="text-align:center">静夜思</h1>
<h3 style="text-align:left">作者：</h3>
<h3 style="text-align:right">李白</h3>
<p style="text-align:justify">
    床前明月光，
    疑是地上霜。
    举头望明月，
    低头思故乡。
    </p>
</body>
```

通过上述代码将一首李白的《静夜思》作为内容进行排版，标题使用 center 属性值进行居中显示。而“作者”则是使用 left 向左对齐，“李白”使用右对齐方式，详细内容则采用两端对齐方式，通过这样的排列我们就按照古诗的排列方式将《静夜思》有规律地排列好了。

text-align 属性设置的对齐方式，不仅可以使标签中的文本内容按照指定的方式对齐。同时还可以对标签中的文本、图片应用对齐方式。下面我们还是通过使用上述诗句排列，详细代码如下所示：

```
<body>
    <h1 style="text-align:center">静夜思</h1>
<h3 style="text-align:left">作者：</h3>
    <img src="libai.jpg"/>
<h3 style="text-align:right">李白</h3>
<p style="text-align:justify">
    床前明月光，
    疑是地上霜。
    举头望明月，
    低头思故乡。
    </p>
</body>
```

text-align 属性只能适用于文本块，而不能直接应用到图片标签中。如果要使用图片同文本一样应用对齐方式，那么必须将图像包含在文本块中。



CSS 只能定义两端对齐方式，并按照要求显示，但对于具体的两端对齐文本如何分配字体空间以实现文本左右两边均对齐，CSS 并不规定，这就需要设计者自行定义了。

5.1.4　基础知识——white-space 属性

white-space 属性就是空白属性，该属性对文本的显示有着重要的影响。在标签上应用 white-space 属性可以影响浏览器对字符串或者文本间空白的处理方式。

在 XHTML 中已经实现了将多余的空白忽略而保留一个空白的方法，代码语法如下所示：

```
<p>窗    内    网    欢    迎    您    ！</p>
```

该文本显示时，会将各文字中间的多个空格缩写为一个空格显示出来，输出在页面的信息如下所示：

```
窗 内 网 欢 迎 您 ！
```

如果要保留空白，则必须使用 CSS 提供的 white-space 属性。该属性有三个可选属性值分别为：pre、nowrap 和 normal。pre 表示文本间的空白不忽略。nowrap 表示文本内容不允许换行，除非使用 hr 元素实现换行。normal 表示按正常显示，文本间多余的空白将被忽略。下面我们通过一段代码来演示如何应用 white-space 属性，详细代码如下所示：

```
<body>
    <h1 style="color:red;text-align:center;white-space:pre">窗    内    网    欢
迎    您    ！</h1>
<div style="white-space:nowrap;text-indent:10mm">
    <p>
窗内网是由郑州汇智计算机技术服务有限公司创建的一个免费视频教程网站，成立于 2002 年。致力于把基于先进信息技术的最佳管理与业务实践普及到客户的管理与业务创新活动中。全面提供企业管理 ERP 软件、服务与解决方案，是中原地区较早的管理软件、客户关系管理软件、小型企业管理软件以及网站建设服务提供商。
    </p>
    <p>
        窗内网结合软件开发和网站建设的实际需求，发布了 Flex、C#、Java、ASP 和 JSP 等系列视频教程。<br>
我们投入大量资源，录制了更多 IT 教学视频教程，包括 SQL Server、PHP、AJAX、Eclipse、Struts、Netbeans、JSF、Hibernate 等开发技术，以及 3ds max、Maya 等三维动画与视频处理教学视频产品等。 早前，公司与电子工业出版社合作编写了《从入门到精通》、《巧学巧用》、《锦绣映像》等系列图书，得到了读者的一致好评和认可！
    </p>
</div>
</body>
```

在上述代码中，元素 h1 中定义 white-space 属性值为 pre 保留了各文字之间的空白，而元素 div 中定义属性值为 nowrap。其中 div 标签中的 p 元素继承了该属性值定义的样式信息。在第一个段落中，由于没有换行标记，文本将以不换行的方式显示整段内容。而第二段文字中使用了
，那么在换行标签处进行换行。

5.1.5 基础知识——ruby-align 属性

ruby-align 属性设置或检索通过 rt 对象指定的注释文本或发音指南的对齐位置。同样该属性有 7 个值，它们分别是：auto、left、center、right、distribute-letter、distribute-space 和 line-edge。

- auto：由浏览器确定对齐方式。对于 ideographic(东亚文本)以 distribute-space 值对齐。对于 Latin 文本则采用 center 值对齐。
- left：根据基本宽度左对齐。

- center：根据基本宽度居中对齐。如果基本宽度小于 ruby 文本的宽度，那么在 ruby 文本的宽度中居中基本宽度。
- right：根据基本宽度右对齐。
- distribute-letter：如果 ruby 文本的宽度小于基本宽度，则 ruby 文本在基本宽度中均匀分布。如果 ruby 文本的宽度大于或等于基本宽度，居中对齐。
- distribute-space：如果 ruby 文本的宽度小于基本宽度，则 ruby 文本在基本宽度中均匀分布。在 ruby 文本中，在第一个字符的前面和最后字符的后面有半个字距的空白区域。如果 ruby 文本的宽度大于或等于基本宽度，居中对齐。
- line-edge：如果 ruby 文本不相邻行边缘，则其被居中。否则 ruby 文本行在基本文本边的上方。

下面通过一段简单的代码来演示，该如何使用 ruby-align 属性和该属性中的一些参数的应用。

```
<body>
<RUBY style="RUBY-ALIGN:distribute-space;">
    <h1 style="ruby-align;left">窗  内  网  欢  迎  您  ！</h1>
<div style=" ruby-align;line-edge">
    <p>
<RT>窗内网是由郑州汇智计算机技术服务有限公司创建的一个免费视频教程网站，成立于 2002 年。致力于把基于先进信息技术的最佳管理与业务实践普及到客户的管理与业务创新活动中。全面提供企业管理 ERP 软件、服务与解决方案，是中原地区较早的管理软件、客户关系管理软件、小型企业管理软件以及网站建设服务提供商。<RT>
    </p>
</div>
</RUBY>
</body>
```

上述代码中我们将标题 ruby-align 样式设置为根据基本宽度左对齐，同时将正文中的 div 设置为 line-edg，如果 ruby 文本不与行边缘相邻，则其被居中。默认情况下该属性的值为 auto。



ruby-align 属性应用于 ruby 标签中，必须和 rt 标签一起使用。否则不会出现想要的效果。

5.1.6　实例描述

通过对一些基础知识的学习，下面我们将通过窗内网图书推荐详细图书页面进行添加样式，将该页面内的文本设置为指定的样式。比如指定图书的名称居中，那么就可以使用 text-align 属性设置，同时还可以使用 white-space 属性去掉多余的空白等功能。

5.1.7　实例应用

【例 5-1】推荐图书详细信息

首先设置显示的图书基本信息，例如图书的名称、编号等一些基本信息。

```
<TABLE border=0 cellSpacing=1 cellPadding=1 width="100%" bgColor=#c6e7ff
align=center>
    <TBODY>
    <TR>
        <TD bgColor=#f3fbff height=25 width=80 align=middle>书名：</TD>
        <TD style="FONT-SIZE: 16px;
FONT-WEIGHT: bold;text-align:left;white-space:pre;white-space:nowrap"
bgColor=#f3fbff colSpan=2 >Eclipse 3.0 从入门到精通</TD>
    <TD bgColor=#f3fbff rowSpan=6 align=middle>
        <DIV class=pic><IMG
        src="images/20090112055110.gif"></DIV></TD></TR>
    <TR>
        <TD bgColor=#f3fbff height=25 align=middle>书号：</TD>
        <TD bgColor=#f3fbff colSpan=2
style="text-align:left">978-7-121-06023-6</TD></TR>
    <TR>
        <TD bgColor=#f3fbff height=25 align=middle>版次：</TD>
        <TD bgColor=#f3fbff colSpan=2
style="text-align:left">2008-05</TD></TR>
    <TR>
        <TD bgColor=#f3fbff height=25 align=middle>页数：</TD>
        <TD bgColor=#f3fbff colSpan=2 style="text-align:left">504</TD></TR>
    <TR>
        <TD bgColor=#f3fbff height=25 align=middle>定价：</TD>
        <TD bgColor=#f3fbff colSpan=2 style="text-align:left">58</TD></TR>
    <TR>
        <TD bgColor=#f3fbff height=25 align=middle>下载：</TD>
        <TD bgColor=#f3fbff align=middle><A
href="http://www.itzcn.com/uploadFile/book/kejian/20090112055110.gif">教学课
件</A>
        </TD>
        <TD bgColor=#f3fbff align=middle><A
href="http://www.itzcn.com/uploadFile/book/anli/20090112055110.gif">素材下载
</A>
        </TD>
    </TR></TBODY></TABLE>
```

在上述代码中我们通过设置 white-space:nowrap 属性避免图书名称换行并去除多余空格，同时使用 text-align:left 属性将文本设置为左边对齐。

接下来则设置“内容要点”文本的样式，详细代码如下所示。

```
<TABLE style="MARGIN-TOP: 8px; BORDER-TOP: #ff8100 3px solid" border=0
cellSpacing=1 cellPadding=0 width="100%" bgColor=#cccccc align=center>
<TBODY>
    <TR>
        <TD style="FONT-SIZE: 16px; FONT-WEIGHT: bold;text-align:left"
bgColor=#ebebeb
height=20> 内 容 要 点 ：</TD></TR>
    <TR>
```

```
        <TD bgColor=#ffffff align=middle>
<TABLE border=0 cellSpacing=0 cellPadding=0 width="96%">
<TBODY>
    <TR>
        <TD></TD></TR>
    <TR>
        <TD>
            <DIV  style="text-indent:10mm;text-align:left;">Eclipse 最初由 OTI
和 IBM 两家公司的 IDE 产品开发组创建。在目前所有的可视化开发工具中，Eclipse 可以说是最有发
展前途的产品之一。本书全面介绍了 Eclipse 的开发调试环境、SWT、JFace、RCP、Eclipse 表单、
Web 开发和插件开发等。
</DIV>
        </TD>
     </TR>
    </TBODY></TABLE>
        </TD>
    </TR>
</TBODY></TABLE>
```

在上述代码中，我们通过设置 div 元素的 text-indent 属性为 10mm 产生段落缩进，设置 text-align 属性为 left 使文本采用向左对齐。

5.1.8　运行结果

通过浏览器运行 book.html 页面将会显示如图 5-1 所示的结果。

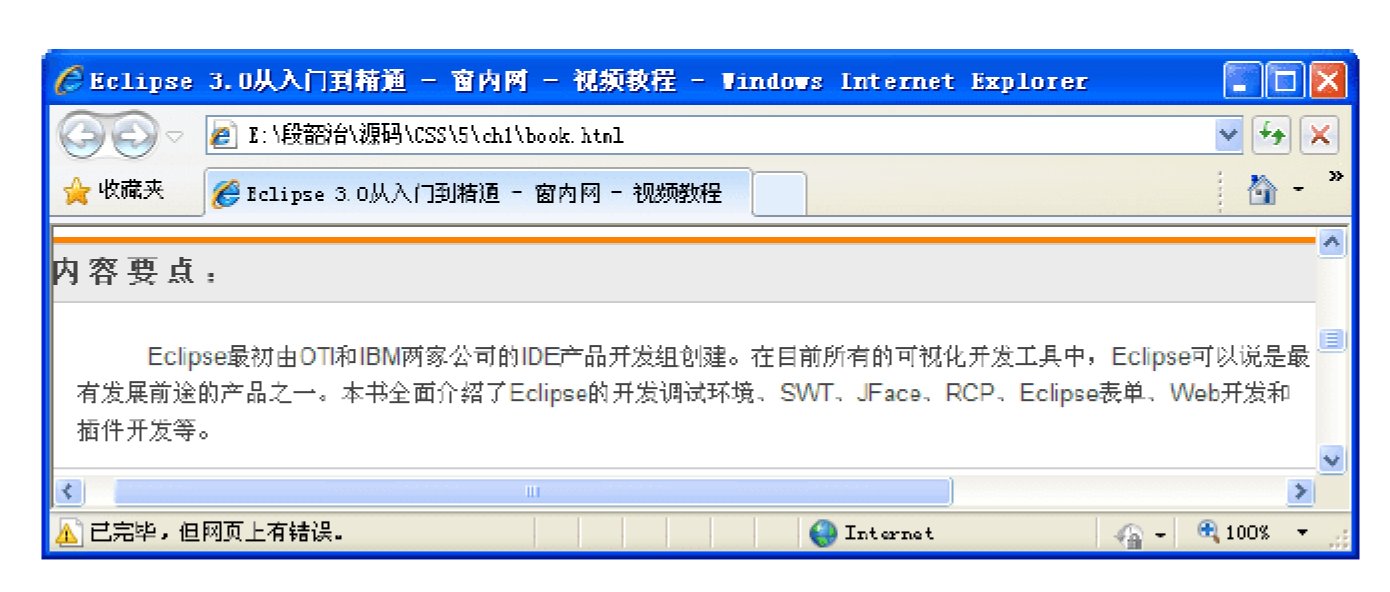


图 5-1　图书推荐详细信息

5.1.9　实例分析



源码解析：

从上面的实例中我们可以看到，通过属性来改变字体文本的对齐方式。在上面讲解到了 text-indent 属性，通过这个属性可以不用在段落前面输入空格，而且缩进的宽度也可以通过数字进行控制，非常方便对页面优化使用。

5.2 网站简介内容

由于各个国家在阅读顺序上可能存在差异，所以有的时候需要能够控制文本的书写方向，或由左到右，或由右到左等。本节将对 direction 属性、layout-flow 属性和 writing-mode 属性的使用进行讲解。

**视频教学：**光盘/视频/5/5.2　网站简介内容.avi　　长度：5 分钟

5.2.1 基础知识——direction 属性

direction 属性用来设置文本流的方向，所谓的文本流就是文本增加的方向。比如古代时期人们喜欢从右向左阅读文章，而现代的人则喜欢从左向右阅读，这个属性就类似于这样的形式，将文字从左向右排序，或者从右向左排序。

该属性有 3 个参数，它们的作用分别是如下所示。

- ltr：文本流从左到右。
- rtl：文本流从右到左。
- inherit：文本流的值不可继承。

以上 3 个参数则是设置文本流的方向和该属性是否被继承等功能。



对于行内元素，只有当 unicode-bidi 属性设置为 embed 或 bidi-override 时才会应用 direction 属性。

在上面提到了 unicode-bidi 属性，direction 属性通常需要和 unicode-bid 属性一起使用才有效。unicode-bidi 属性允许同一个页面里存在从不同方向读进的文本，同样它也有 3 个属性，如下所示。

- normal：对行内文本进行重新排序，使文本严格按照从左到右的顺序排列。
- bidi-override：按照 direction 属性的值进行排序。
- embed：效果同 normal。

unicode-bidi 属性的主要作用就是和 direction 属性一起使用，接下来通过一段代码来讲解如何使用 direction 属性和 unicode-bidi 属性改变一段文本流的方向，详细代码如下所示。

```
<DIV  style="unicode-bidi: bidi-override; direction:rtl">Eclipse 最初由 OTI 和
IBM 两家公司的 IDE 产品开发组创建。在目前所有的可视化开发工具中，Eclipse 可以说是最有发展
前途的产品之一。本书全面介绍了 Eclipse 的开发调试环境、SWT、JFace、RCP、Eclipse 表单、
Web 开发和插件开发等。
</DIV>
```

上述代码中通过设置 unicode-bidi 属性值为 bidi-override 和设置 direction 值为 rtl 让文本流从右到左进行排序。

5.2.2　基础知识——layout-flow 属性

layout-flow 属性用来设置或检索对象内文本的流动和方向，同样它也是从右向左流动的。唯一和上面讲解到的 direction 属性不同的是，它以上下排列字体形式呈现。一般文本内容不是从左向右就是从右向左，而 layout-flow 属性则是按照从上向下的排列方式将文本呈现出来。同样该属性也可以做到自左边流入，该属性有 horizontal 和 vertical-ideographic 两个值，它们的作用简介如下。

- horizontal：对象中的内容自左边流入。下一行在前一行下面。这个值适于拉丁语系。
- vertical-ideographic：对象中的内容自上而下流入，下一行在前一行左面。这个值适于亚洲语系。



当 layout-flow 属性值发生变化时，text-align 属性与 vertical-align 属性的作用也将发生变化。

上面讲述的是 text-align 属性的设置方法，以及检索对象中文本的对齐方式。下面通过一段代码来讲解如何使用 text-align 属性，详细代码如下所示。

```
<DIV  style="layout-flow:vertical-ideographic; height:100px">Eclipse 最初由
OTI 和 IBM 两家公司的 IDE 产品开发组创建。在目前所有的可视化开发工具中，Eclipse 可以说是最
有发展前途的产品之一。本书全面介绍了 Eclipse 的开发调试环境、SWT、JFace、RCP、Eclipse
表单、Web 开发和插件开发等。
</DIV>
```

通过上述代码可以发现，在 style 属性中指定了 layout-flow 属性值为 vertical-ideographic，也就是将对象中的内容设置为自上而下流入。

5.2.3　基础知识——writing-mode 属性

writing-mode 属性用来设置或检索对象的内容块固有的书写方向。例如，西方语言默认采用的是从左向右、从上向下的书写方式。但是亚洲语言通常采用从上向下、从右向左的书写方式。因此为了方便不同阅读习惯的人查看文本，可以使用该属性限制文本显示。该属性有两个参数值如下所示。

- lr-tb：从左向右，从上到下的横向排列。
- tb-rl：从上到下，从右向左的纵向排列。

该属性和 layout-flow 属性运行效果相同，下面还通过一段代码来讲解如何使用该属性完成文本样式，详细代码如下所示。

```
<DIV  style="writing-mode:tb-rl ; height:100px;text-align:left">
Eclipse 最初由 OTI 和 IBM 两家公司的 IDE 产品开发组创建。在目前所有的可视化开发工具中，
Eclipse 可以说是最有发展前途的产品之一。本书全面介绍了 Eclipse 的开发调试环境、SWT、JFace、
RCP、Eclipse 表单、Web 开发和插件开发等。
</DIV>
```

在上述代码中，首先设置了 writing-mode 属性值为 tb-rl，使内容从上向下流动，字体从右向左排序。接下来设置 text-align 的值为 left，使内容左对齐。

5.2.4 实例描述

在讲了这么多关于文本内容流的方向属性后，下面进入开发模式。通过一个实例的制作，让大家更好地了解和应用这些属性，从而做到恰到好处。本节仍然通过窗内网网站介绍进行讲解，通过设置网站介绍的内容样式让内容文本以从上到下的方式显示。

5.2.5 实例应用

【例 5-2】网站简介内容

首先在页面内定义一个 p 元素，并将该元素中的 text-align 属性设置为 left，同时设置 writing-mode 为 tb-rl。

```
<DT>网站介绍 </DT>
    <DD class=about>
    <P  style="text-align:left;writing-mode:tb-rl ; height:300px">
nbsp;   郑州汇智计算机技术服务有限公司成立于 2002 年。致力于把基于先进信息技术的最佳管理与业务实践普及到客户的管理与业务创新活动中，全面提供企业管理 ERP 软件、服务与解决方案，是中原地区较早的管理软件、客户关系管理软件、小型企业管理软件以及网站建设服务提供商    汇智科技公司是北京新网郑州地区核心代理服务商，提供域名注册、服务器托管和虚拟主机服务。公司关注 IT 教育培训市场的发展和变化，致力于网络在线教育，推进无纸化教学进程    窗内网(www.itzcn.com)结合软件开发和网站建设的实际需求，发布了 Flex、C#、Java、ASP.NET、ASP 和 JSP 等系列视频教程。我们投入大量资源，录制了更多 IT 教学视频产品，包括 SQL Server、VB.NET、PHP、AJAX、Eclipse、Struts、Netbeans、JSF、Hibernate 等开发技术，以及 3ds max、Maya 等三维动画与视频处理教学视频产品    早前，公司与电子工业出版社合作编写了从入门到精通，巧学巧用，锦绣映像等系列图书，得到了读者的一致好评和认可。
</P>
    </DD>
```

通过上面的实例可以知道，使用 text-align 属性可以设置文本内容排列的方向，即向左排列还是向右排列。设置 writing-mode 则是让网站介绍的内容采用从上向下的流动方式。

5.2.6 运行结果

保存页面，运行 aboutus.html，页面运行结果如图 5-2 所示。

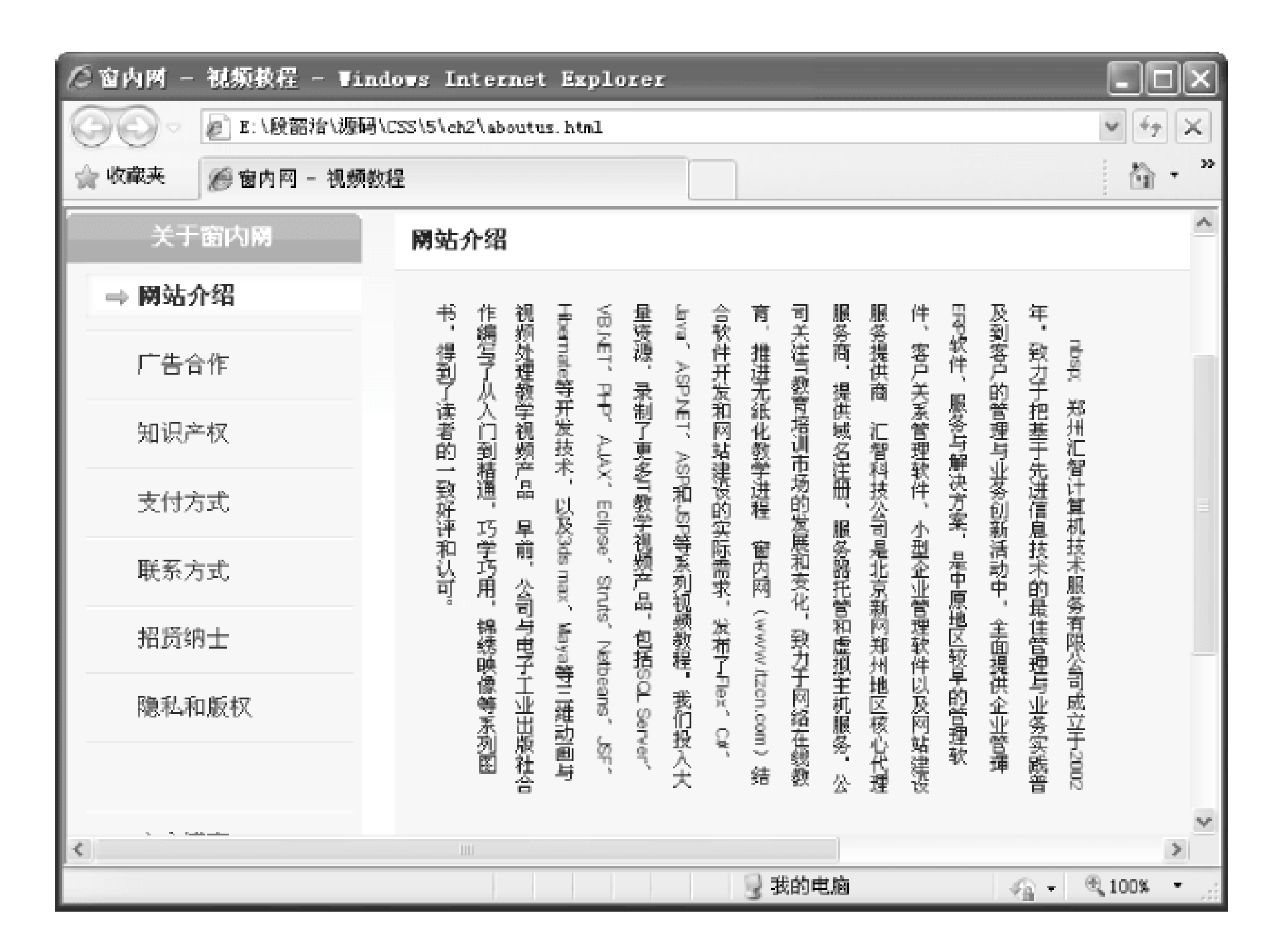


图 5-2　网站简介

5.2.7　实例分析



源码解析：

该实例中，使用 writing-mode 属性设置文本内容块固有的书写方向。在这里设置该属性值为 tb-rl，表示使内容以从上向下流的形式显示。同时该属性和 text-align 属性一起使用，从而完成文本内容的排列方式。这样就很简单地完成了网站介绍模块的显示了。

5.3　详细新闻信息内容

本节主要讲解文本的排列方式。所谓的排列方式就是所需要显示的文本内容以一种显示方式排列在页面内。比如文本的换行方式，可以通过设置属性使文本的换行方式以日文文本的换行规则来显示。下面则对 text-autospace 属性、text-kashida-space 属性、word-break 属性、line-break 属性和 word-wrap 属性进行讲解。



视频教学：光盘/视频/5/5.3　详细新闻信息内容.avi　　　长度：6 分钟

5.3.1　基础知识——text-autospace 属性

text-autospace 属性用来设置或检索对象文本的自动空格和紧缩空格宽度调整的方式。比如设置当空格表意字符时，扩展空格的宽度或者在表意字和数字字符之间创建额外空格等。该属性有 5 个参数，它们的功能简介如下。

- none：无调整发生。

- ideograph-alpha：在表意字和非表意字之间创建额外空格。
- ideograph-numeric：在表意字和数字字符之间创建额外空格。
- ideograph-parenthesis：在常规(非宽度)插入语和表意字之间创建额外空格。
- ideograph-space：当空格相邻表意字符时，扩展空格的宽度。

text-autospace 属性默认情况下值为 none，即无调整状态，没有任何变化。下面通过一段代码介绍如何使用 text-autospace 属性设置文本的自动空格和紧缩空格宽度的调整，详细代码如下所示。

```
<P style="text-autospace: ideograph-space">郑州汇智计算机技术服务有限公司成立于2002年。致力于把基于先进信息技术的最佳管理与业务实践普及到客户的管理与业务创新活动中，全面提供企业管理ERP软件、服务与解决方案，是中原地区较早的管理软件、客户关系管理软件、小型企业管理软件以及网站建设服务提供商    汇智科技公司是北京新网郑州地区核心代理服务商，提供域名注册、服务器托管和虚拟主机服务。</P>
```

上述代码中，设置 text-autospace 属性值为 ideograph-space 时用来设置空格相邻表意字符时扩展空格的宽度。

5.3.2 基础知识——text-kashida-space 属性

text-kashida-space 属性用来设置或检索如何拉伸字符来调节文本行排列。该属性只有一个参数值为 length，用来标识 kashida 膨胀与空格膨胀的比例。当该数值为 100%时仅有 kashida 膨胀。当该数值为 0%时仅有空格膨胀。默认值为 0%。

该属性使用语法如下所示。

```
<P style="text-justify:distribute;text-kashida-space:50%;">郑州汇智计算机技术服务有限公司成立于2002年。致力于把基于先进信息技术的最佳管理与业务实践普及到客户的管理与业务创新活动中，全面提供企业管理ERP软件、服务与解决方案，是中原地区较早的管理软件、客户关系管理软件、小型企业管理软件以及网站建设服务提供商    汇智科技公司是北京新网郑州地区核心代理服务商，提供域名注册、服务器托管和虚拟主机服务。</P>
```

text-kashida-space 属性和 text-justify 属性一起使用，text-justify 属性用来设置或检索对象内文本的对齐方式，它有 6 个参数。

- auto：允许浏览器用户确定使用的两端对齐法则。
- inter-word：通过增加字之间的空格对齐文本。该行为是对齐所有文本行最快的方法。它的两端对齐行为对段落的最后一行无效。
- newspaper：通过增加或减少字或字母之间的空格对齐文本。适用于拉丁文字母表两端对齐的最精确格式。
- distribute：处理空格很像 newspaper，适用于东亚文档。
- distribute-all-lines：两端对齐行的方式与 distribute 相同，也同样不包含两段对齐段落的最后一行。适用于表意字文档。
- inter-ideograph：为表意字文本提供两端对齐。它增加或减少表意字和词间的空格。

5.3.3　基础知识——word-break 属性

word-break 属性用于设置或检索对象内文本的字内换行。尤其在出现多种语言时。通过设置该属性来修改文本显示规则，该属性有 3 个参数。

- normal：默认值。允许在非亚洲语言中的单词之间换行，并允许在亚洲语言的任意两个字之间换行。
- break-all：允许在非亚洲语言中的单词内的任意两个字母中间换行。
- keep-all：不允许在亚洲语言的一个单词内的几个字之间换行。

在下面代码中，通过设置不同的 word-break 属性值，显示文本内容不同的行为，详细代码如下所示：

```
<P  style="word-break:keep-all">郑州汇智计算机技术服务有限公司成立于 2002 年。致力于把基于先进信息技术的最佳管理与业务实践普及到客户的管理与业务创新活动中，全面提供企业管理 ERP 软件、服务与解决方案，是中原地区较早的管理软件、客户关系管理软件、小型企业管理软件以及网站建设服务提供商    汇智科技公司是北京新网郑州地区核心代理服务商，提供域名注册、服务器托管和虚拟主机服务。 </P>
<hr/>
<P style="word-break:break-all">公司关注 IT 教育培训市场的发展和变化，致力于网络在线教育，推进无纸化教学进程    窗内网(www.itzcn.com)结合软件开发和网站建设的实际需求，发布了 Flex、C#、Java、ASP.NET、ASP 和 JSP 等系列视频教程。我们投入大量资源，录制了更多 IT 教学视频产品，包括 SQL Server、VB.NET、PHP、AJAX、Eclipse、Struts、Netbeans、JSF、Hibernate 等开发技术，以及 3ds max、Maya 等三维动画与视频处理教学视频产品    早前，公司与电子工业出版社合作编写了从入门到精通，巧学巧用，锦绣映像等系列图书，得到了读者的一致好评和认可。
</P>
```

上述代码中，使用 hr 元素显示一条线段，将内容分为上下两部分。其中，上部分将 word-break 属性值设置为 keep-all 使文本依照亚洲语言的 normal 方式显示，下面部分将 word-break 属性值设置为 break-all，则是所有的内容与所有非亚洲语言的 normal 方式显示。

5.3.4　基础知识——line-break 属性

line-break 属性用来设置或检索日文文本的换行规则。其作用是以日文文本的换行规则将文本内容呈现在页面上，该属性有两个参数分别如下所示。

- normal：应用日文文本的默认换行规则。
- strict：强制日文文本换行规则的严谨性。

下面代码中演示如何使用该属性对日文文本进行格式化，详细代码如下所示：

```
<P style="line-break:strict"> </P>
```

5.3.5　基础知识——word-wrap 属性

word-wrap 属性用于设置或检索当前行超过指定容器边界时是否自动换行。也就是说如果

定义的文本在指定的容器内显示时，如果文本宽度大于容器的宽度时是否进行换行显示。该属性同样有两个值，详细如下所示。

- normal：允许内容超出指定的容器边界。
- break-word：内容将在边界内换行。

该属性的两个参数是用来控制容器内部内容的显示方式、是否换行等功能。下面通过一个简单的实例，比较两个参数的区别，详细代码如下所示：

```
<div style=" width:200px;word-wrap:break-word">郑州汇智计算机技术服务有限公司成立于 2002 年。致力于把基于先进信息技术的最佳管理与业务实践普及到客户的管理与业务创新活动中，全面提供全面提供企业管理 ERP 软件、服务与解决方案，是中原地区较早的管理软件、客户关系管理软件、小型企业管理软件以及网站建设服务提供商    汇智科技公司是北京新网郑州地区核心代理服务商，提供域名注册、服务器托管和虚拟主机服务。
</div>
```

上述代码中，当内容的宽度大于定义 div 元素的宽度时，将会根据 word-wrap 属性指定的值进行显示。如果该参数取值为 normal，那么容器内的内容将不会换行而是将容器撑开显示。如果取值为 break-word，那么内容则会进行换行显示。

5.3.6 实例描述

本实例通过新闻详细信息页面的文本显示样式，深入讲解学习到的一些文本排列方式的样式应用。通过设置新闻的详细内容，讲解自动换行、换行是否截取、如何拉伸字符来调节文本行排列等功能的实现，详细步骤如下所示。

5.3.7 实例应用

【例 5-3】新闻详细内容

首先需要布局页面新闻显示内容，将需要显示的新闻内容写入 div 元素中，通过设置 div 元素样式来完成整个新闻详细内容的样式排序，详细代码如下所示：

```
<TBODY>
        <TR>
            <TD style="LINE-HEIGHT: 140%; FONT-SIZE: 9pt"
id=dede_content width=659>
                <div class="mscss" style="">中新网 8 月 6 日电
受近日大盘震荡整理走势影响，沪深两市双双于 10 日均线下方微幅低开。开盘后指数掉头向上再试 10 日均线，但在短期走弱的预期之下，大盘上冲乏力，目前沪指下跌 0.34%。板块方面，创业板大涨，领涨大盘。盘面上看，农牧饲渔板块再度大涨 2.97%，暂时领涨两市，木业家具、食品、物流、券商信托等板块涨势居前。目前民航机场、家电、房地产、石油、汽车等权重板块位居跌幅前列。个股方面，目前两市逾 1400 家个股下跌，新股 N 国腾大涨 106.50%，另两家新股 N 双林和 N 尤洛卡分别上涨 81.44% 和 59.98%。普利特、四川圣达、成飞集成、新农开发、亚盛集团纷纷封涨停。两市暂无跌停个股，滨江集团大跌 6.86%居首，信邦制药、空港股份、九龙山、苏泊尔位居跌幅前列。
截至发稿，沪指报 2615.43 点，跌 5.03 点，跌幅 0.19%，成交量 208.5 亿元；深成指报 10715.3 点，跌 52.73 点，跌幅 0.49%，成交量 203.8 亿元。
```

```
(责任编辑：李立婷)
            </div>
        </TD>
    </TR>
</TBODY>
```

在上述代码中，定义 div 元素显示新闻详细内容。通过设置 div 元素的 class 属性来指定样式，下面则是 div 元素内文本样式信息，详细代码如下：

```
<STYLE type=text/css>
.mscss{
    text-autospace: ideograph-space; /*设置或检索对象文本的自动空格和紧缩空格宽度调整的方式。*/
    text-justify:distribute;text-kashida-space:50%;/*设置或检索如何拉伸字符来调节文本行排列*/
    word-break:normal;       /*设置或检索如何拉伸字符来调节文本行排列*/
    width:640px;             /*设置 div 的宽度*/
    word-wrap:break-word;    /*换行是否断开*/
    text-indent:6mm;         /*自动缩进*/
    }
</STYLE>
```

上述代码则是设置 div 元素的样式代码，通过设置 text-autospace 属性使文本自动调整空格和紧缩空格的宽度。同时还设置了 text-kashida-space 属性，使文本拉伸字符调整文本行排列等功能属性。

5.3.8　运行结果

保存修改后的页面，在浏览器中运行 NewMessage.htm 页面。显示如图 5-3 所示的效果。



图 5-3　新闻详细信息页面

5.3.9 实例分析



源码解析：

在本实例中，修改 text-indent 属性使段落缩进 6mm。修改 word-wrap 属性为 break-word，使文本内容大于容器的边界时自动换行；修改 text-autospace 属性设置紧缩空格的宽度；将 word-break 属性设置为 normal，控制文本中单词之间的换行。

5.4 新闻信息列表设置

本节将对如何设置对象中的文本的字体大小写、文本修饰属性以及对多余字符的截取等功能进行讲解，它们分别对应的则是 text-transform 属性、text-decoration 属性和 text-overflow 属性。



视频教学：光盘/视频/5/5.4 新闻信息列表设置.avi 长度：5 分钟

5.4.1 基础知识——text-transform 属性

text-transform 属性用于检索或设置对象中的文本的大小写，并将文本的内容由小写转换为大写或者由大写转换为小写。该属性有 4 个值，它们的作用就是用来控制定义显示内容的显示格式，下面则是该属性值的功能。

- none：无转换发生。
- capitalize：将每个单词的第一个字母转换成大写，其余无转换发生。
- uppercase：转换成大写。
- lowercase：转换成小写。

通过下面一段代码将页面上显示的内容进行大小写转换。通过设置 text-transform 属性配置它的值就可以完成，该属性的默认值为 none，没有任何转换效果，详细代码如下所示：

```
<div  style="text-transform:capitalize">
    hello world
</div>
<div  style="text-transform:uppercase">
    hello world
</div>
<div  style="text-transform:lowercase">
    HELLO WORLD
</div>
```

在上述代码中定义了 3 个 div 元素，在每个 div 中都有英语字母，第一个 div 中写着“hello world”，第二个 div 和第一个 div 相同，唯一不同的就是第一个 div 的 text-transform 属性值为

capitalize，从而将单词的首字母大写。第二个 div 的 text-transform 属性则是将所有小写字母全部转换为大写字母。第三个 div 定义的 text-transform 属性则是将所有大写字母转换为小写，因此当运行页面时呈现在页面上的则是第一个 div 为“Hello World”，第二个 div 显示的内容为“HELLO WORLD”，第三个 div 呈现在页面上则是“hello world”。

5.4.2　基础知识——text-decoration 属性

text-decoration 属性是文本修饰属性，该属性可以为页面提供多种文本修饰效果，比如下划线、删除线和闪烁等。该属性值有 5 个参数，它们的作用分别如下所示。

- none：无装饰。
- blink：闪烁。
- underline：下划线。
- line-through：贯穿线。
- overline：上划线。

text-decoration 属性的默认值非常特殊，在不同的对象下默认是不同的。没有文本的对象，此属性不会起作用。

```
<div>
    <p id="p1"style="text-decoration:underline">郑州汇智计算机技术服务有限公司成立于 2002 年。</p>
    <p id="p2"style="text-decoration:overline">致力于把基于先进信息技术的最佳管理与业务实践普及到客户的管理与业务创新活动中，全面提供企业管理 ERP 软件、服务与解决方案，是中原地区较早的管理软件、客户关系管理软件、小型企业管理软件以及网站建设服务提供商</p>
    <p p="3" style="text-decoration:line-through">汇智科技公司是北京新网郑州地区核心代理服务商，提供域名注册、服务器托管和虚拟主机服务。</p>
<p id="p4" style="text-decoration:blink">窗内网</p>
</div>
```

在上述代码中，通过设置 p 元素的 text-decoration 属性修饰不同的文本内容，其中 p1 元素内显示着下划线内容，p2 元素内显示着上划线内容，p3 则显示删除线也就是贯穿线内容，p4 则是显示闪烁内容。



只有 Firefox、Mozilla 和 Netscape 浏览器支持 blink 闪烁效果，而 IE 和其他浏览器(如 Opera)都不支持该效果。

5.4.3　基础知识——text-overflow 属性

text-overflow 属性是一个比较特殊的样式，我们可以用它代替通常所用的标题截取函数，而且这样做对搜索引擎更加友好。例如：标题文件有 50 个汉字，而我们的列表可能只有 300px 的宽度。如果用标题截取函数，则标题不是完整的，如果我们用 CSS 样式 text-overflow:ellipsis，输出的标题是完整的，只是受容器大小的局限不显示出来罢了。

text-overflow 属性用来设置或检索是否使用一个省略标记标示对象内文本的溢出，下面是 text-overflow 属性值简介。

- clip：不显示省略标记，而是简单地裁切。
- ellipsis：当对象内文本溢出时显示省略标记。

text-overflow 属性仅是注解，用来确定当文本溢出时是否显示省略标记。要想实现溢出时产生省略号的效果，还必须强制文本在一行内显示(white-space:nowrap)及溢出内容为隐藏(overflow:hidden)。下面通过一个简单的实例来实现如何让 div 溢出内容显示省略号效果，详细代码如下所示：

```
<div style="width:300px; line-height:25px; text-overflow:ellipsis;
white-space:nowrap; overflow:hidden;">任何使用窗内(http://www.itzcn.com)的用户
均应仔细阅读本声明，用户可选择不使用窗内网，用户使用窗内的行为将被视为对本声明全部内容的认
可！</div>
</div>
```

在上述代码中，首先设置 div 元素的宽度，当内容超过定义 div 元素设置的宽度时，div 元素后面显示内容以省略号代替；text-overflow 属性设置为 ellipsis 则应当显示省略号功能，如果设置为 clip 则直接截断不显示省略号。

5.4.4 实例描述

讲到这里大家一定对上面介绍的这些属性非常熟悉了，下面通过一个实例让大家深入地了解和更加熟练地使用这些属性。这个实例是通过一个简单的新闻列表进行讲解，实例要求新闻列表标题字体加粗，而且使用下划线样式，而标题下面的简介则需要缩进，当新闻简介内容大于定义存放文本的容器时，就会将简介内容后面部分用省略号代替。

5.4.5 实例应用

【例 5-4】新闻列表页面

首先在页面上定义 div 元素，在每个 div 元素中写入要显示的新闻列表的标题和新闻简介信息，详细代码如下所示：

```
<TABLE border=0 width="100%">
        <TBODY>
            <TR>
            <TD style="FONT-SIZE: 14px"
                width="100%">    <A
                title=""

    href="http://localhost:8080/News/userinfo.do?o=newsmsbyid&amp;id=119"><
STRONG>冒名顶替事件引发高校清查
                1 所学院清退 300 多人</STRONG></A></TD></TR>
                <TR>
            <TD>  日期: 2010-08-06 16:35:49 单击: <FONT
                color=#ff0000>2 </FONT>  </TD></TR>
            <TR>
```

```
            <TD><div>
        6 月初的一天，大三学生“张阳"被辅导员叫到办公室，辅导员直截了当地告诉他：“上面查
出你是假学生，你得退学。"
        “张阳"似乎有所准备，他站着，低头沉默了一会儿，问“还有没有余地"？辅导员无奈地摇
头：“如果你还想听课，可以办一个旁听手续。"
</div></TD></TR></TBODY></TABLE>
```

上述代码则是定义了新闻的标题和新闻内容的简介信息，同时设置了 div 容器。下面则是设置新闻简介内容样式(div 元素)，当内容大于容器的宽度时，将切断后面的内容信息而以省略号代替，同时设置文本内容自动缩进样式，详细代码如下所示：

```
    div{
    width:650px;
        line-height:25px;
        text-overflow:ellipsis;
        white-space:nowrap;
        overflow:hidden;
        }
```

代码中，使用 width 属性设置 div 容器的大小，当内容大于容器的大小时，则设置 text-overflow 属性值为 ellipsis 来截取文本。接下来则是设置标题的下划线样式，详细代码如下所示：

```
STRONG{
    text-decoration:underline
    }
```

通过设置 text-decoration 属性值为 underline，就可以实现为标题内容添加下划线样式，同样也可以设置为上划线或者删除线(贯穿线)。

5.4.6　运行结果

在浏览器中运行项目中的 newlist.html 页面，运行结果如图 5-4 所示。



图 5-4　新闻列表页面

5.4.7 实例分析


源码解析：

在上面的实例中，通过设置 CSS 样式让新闻列表内容显示得更加美观。首先使用 text-decoration 属性来定义标题显示下划线样式，然后通过设置 text-overflow 属性截取容器内过多的文本信息并用省略号代替。同时设置文本自动缩进功能。

5.5 常见问题解答

5.5.1 CSS 闪烁效果问题



使用 CSS 实现闪烁效果失败，没有显示出预期的结果？

网络课堂：http://bbs.itzcn.com/thread-11370-1-1.html

学到这里大家应该对 text-decoration 属性已经非常了解了，也能够使用该属性实现效果。可是今天遇到了一个很头痛的问题，我使用 text-decoration 属性并指定它的值为 blink，实现一个闪烁效果。当我运行该页面时出现一个很严重的问题，没有闪烁的效果，很不理解，代码如下所示：

```
<div id="p4" style="text-decoration:blink">
<a href="http://www.itzcn.com/msg">(您有新消息)</a>
</div>
```

上述代码是当有系统信息时，执行这段代码，使 a 元素超链接不断闪烁提示用户有新消息。可是当我运行时却没有在我意料之中闪烁，请求帮忙！

【解决办法】从代码上来看，属性使用是正确的，之所以造成不会出现闪烁效果的主要原因是浏览器不支持，blink 闪烁效果只能在 Firefox、Mozilla 和 Netscape 浏览器中被支持，而 IE 和其他浏览器(如 Opera)都不支持该效果。

5.5.2 ruby-align 属性无效



使用 text-decoration 属性时运行没有效果？

网络课堂：http://bbs.itzcn.com/thread-11371-1-1.html

本人在做实例的时候使用 ruby-align 属性，如果 ruby 文本与行边缘不相邻，则其被居中显示；否则 ruby 文本行在基本文本边的上方。可是我把内容放在页面上运行时，却没有任何效果，而且和普通的段落没有任何差别，代码如下所示：

```
<div style="ruby-align;line-edge">
    <p>
简介：窗内网 C#2008 视频教程系统讲解了 C#基础知识，并通过大量的案例引导学生循序渐进地深入掌握 C#开发实际项目，为方便学生动手操作，每个视频界面提供案例下载，学生在观看视频的同时，如果对该案例有兴趣，可以下载到 ... 视频教程。
    </p>
</div>
```

上述代码中用来介绍一本书的内容简介，可是当显示在页面中时却不是想要的效果!

【解决办法】ruby-align 属性应用于 ruby 标签中，必须和 rt 标签一起使用。否则不会出现想要的效果。

5.5.3　显示省略号问题



使用 text-overflow 属性截取时内容显示省略号的问题。

网络课堂：http://bbs.itzcn.com/thread-11372-1-1.html

今天使用 CSS 中的 ext-overflow 属性，实现当内容过多时溢出容器，则截取显示省略号。可是当我将内容显示到页面时，页面却被定义内容过多的容器“撑坏”了，没有实现自动截取过多字符串用省略号代替的效果，代码如下所示，大家帮我看看怎么个情况。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>52CSS</title>
<style type="text/css">
li { line-height:25px; text-overflow:ellipsis; white-space:nowrap;
overflow:hidden;}
</style>
</head>
<body>
<ul>
<li><a href="http://ww.itzcn.com">窗内网免费视频教程 </a>
<li><a href="http://ww.itzcn.com">窗内网免费视频教程 </a>
<li><a href="http://ww.itzcn.com">窗内网免费视频教程 </a>
<li><a href="http://ww.itzcn.com">窗内网免费视频教程 </a>
<li><a href="http://ww.itzcn.com">窗内网免费视频教程 </a>
<li><a href="http://ww.itzcn.com">窗内网免费视频教程 </a>
</ul>
</body>
</html>
```

【解决办法】从代码上看，应该是定义的 li 没有宽度，也就是说当该属性运行时找不到自己的标准，所以没有任何效果。为它指定一个标准宽度后，它就会依照这个宽度进行截取文本，并以省略号代替。

5.6 习　　题

1. 填空题

(1) __________用来设置文本的水平对齐方式。

(2) 使用 ruby-align 属性时，需要和__________元素配合使用。

(3) layout-flow 属性用来设置或检索对象内文本的流动和方向，该属性有 horizontal 和__________属性值。

(4) 在 CSS 样式中，________属性用来设置文本的闪烁功能，而且在 IE 浏览器中不支持。

(5) 通常使用 text-transform 属性，设置文本大小写转换，当需要将小写字母转换为大写字母时该属性的值为__________。

(6) word-wrap 属性中 break-word 值的作用是__________。

2. 选择题

(1) 下面选项中不属于 direction 属性的值是__________。

A. ltr　　B. rtl　　C. inherit　　D. bidi

(2) __________属性就是空白属性，该属性对文本的显示有着重要的影响。

A. word-wrap　　B. word-break　　C. white-space　　D. line-break

(3) text-autospace 属性用来设置或检索对象文本的自动空格和紧缩空格宽度调整的方式，下面选项中不属于该属性的值的是__________。

A. ideograph-alpha　　B. ellipsis

C. ideograph-parenthesis　　D. ideograph-space

(4) 如果我们需要在文本中显示一条删除线(贯穿线)，那么就需要指定 text-decoration 属性的__________值。

A. blink　　B. underline　　C. line-through　　D. overline

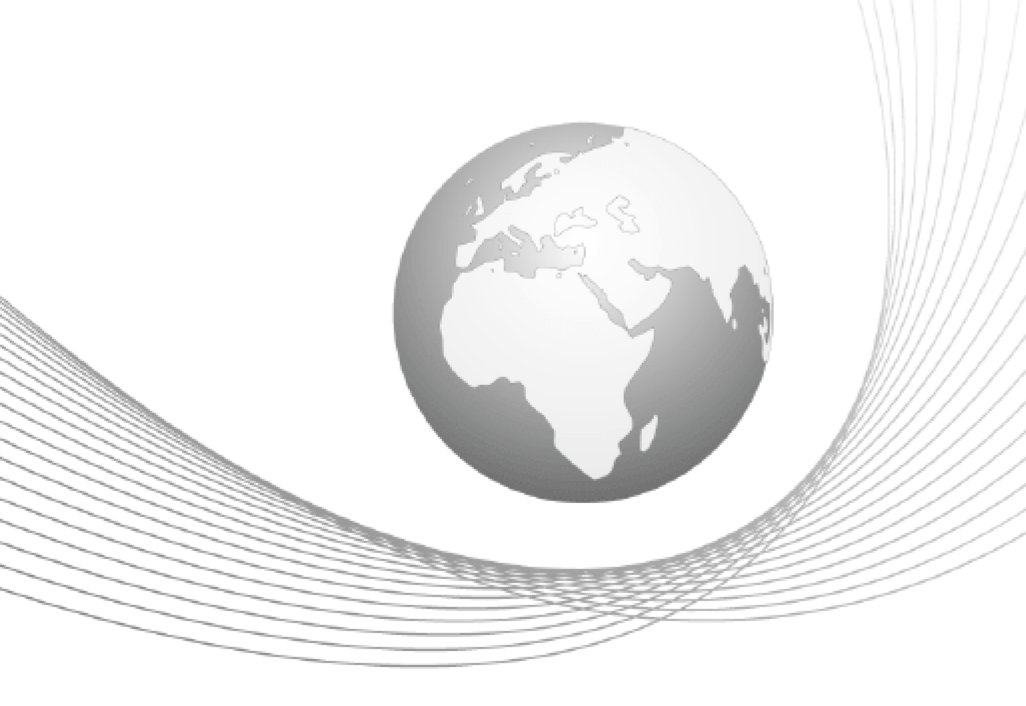
3. 上机练习

上机练习：视频信息列表页面。

实现窗内网视频列表显示模块，该模块要求标题加粗，而且具有超链接功能，标题下面简介要求使用小号字体，文本流动方向由左向右，当内容宽度大于定义容器的大小后将以省略号代替。运行结果如图 5-5 所示。



图 5-5　视频列表页面



第 6 章　边框与尺寸属性

内容摘要：

在 HTML 中，我们使用表格来创建文本周围的边框；但如果使用 CSS 边框属性，我们就可以创建出效果出色的边框，并且可以应用于任何元素。元素外边距内就是元素的边框(border)。

元素的边框就是围绕元素内容和内边距的一条或多条线。每个边框都有 3 个属性：宽度、样式以及颜色。本章我们就来详细介绍设置边框以及边框尺寸的各种属性，通过灵活运用这些属性来为页面元素设置漂亮的边框样式。

学习目标：

- 掌握边框样式属性 border-style、border-width、border-color 和 border
- 熟悉边框各个局部样式属性
- 掌握设置边框尺寸属性

6.1 设置一个彩色的盒子

页面元素的边框就是将元素内容及间隙包含在其中的边线，和表格的外边线非常相似。一个页面元素的边框可以从宽度、样式和颜色三方面来描述，这三个方面决定了边框所显示出来的外观。CSS 提供了 border-style、border-width 和 border-color 三个属性来设定边框的这三个方面。

**视频教学：**光盘/视频/6/6.1 设置一个彩色的盒子.avi 长度：5 分钟

6.1.1 基础知识——border-style 属性

border-style 属性用于设置页面元素边框的样式，也就是风格。样式是边框最重要的一个方面，因为样式控制着边框的显示，没有样式，将根本没有边框。CSS 提供了 10 种边框样式，如表 6-1 所示。

表 6-1 边框样式

样式取值	说 明
none	默认值。无边框。不受任何指定
hidden	隐藏边框。IE 不支持该样式
dotted	在 Mac 平台上 IE4+与 Windows 和 UNIX 平台上 IE5.5+为点线。否则为实线
dashed	在 Mac 平台上 IE4+与 Windows 和 UNIX 平台上 IE5.5+为虚线。否则为实线
solid	实线边框
double	双线边框。两条单线与其间隔的和等于指定的 border-width 值
groove	根据 border-color 的值画 3D 凹槽线式边框
ridge	根据 border-color 的值画 3D 凸槽线式边框
inset	根据 border-color 的值画 3D 内嵌效果的边框
outset	根据 border-color 的值画 3D 凸起效果的边框

border-style 属性的语法格式：

```
border-style : none | hidden | dotted | dashed | solid | double | groove | ridge
| inset | outset
```

下面来设置一个 div 元素的边框样式，CSS 样式代码如下所示：

```
div.div1 {
    border-style: none;     //默认值，无边框
}
div.div2 {
    border-style: hidden;   //隐藏边框
}
```

```
div.div3 {
border-style: solid;        //实线边框
}
div.div4 {
    border-style: double;   //双线边框
}
div.div5 {
    border-style: groove;   //凹槽线式边框
}
div.div6 {
    border-style: ridge;    //凸槽线式边框
}
div.div7 {
    border-style: inset;    //内嵌效果的边框
}
div.div8 {
    border-style: outset;   //凸起效果的边框
}
div.div9 {
    border-style: dashed;
}
div.div10 {
    border-style: dotted;
}
```

上述各个边框样式的执行效果如图 6-1 所示。

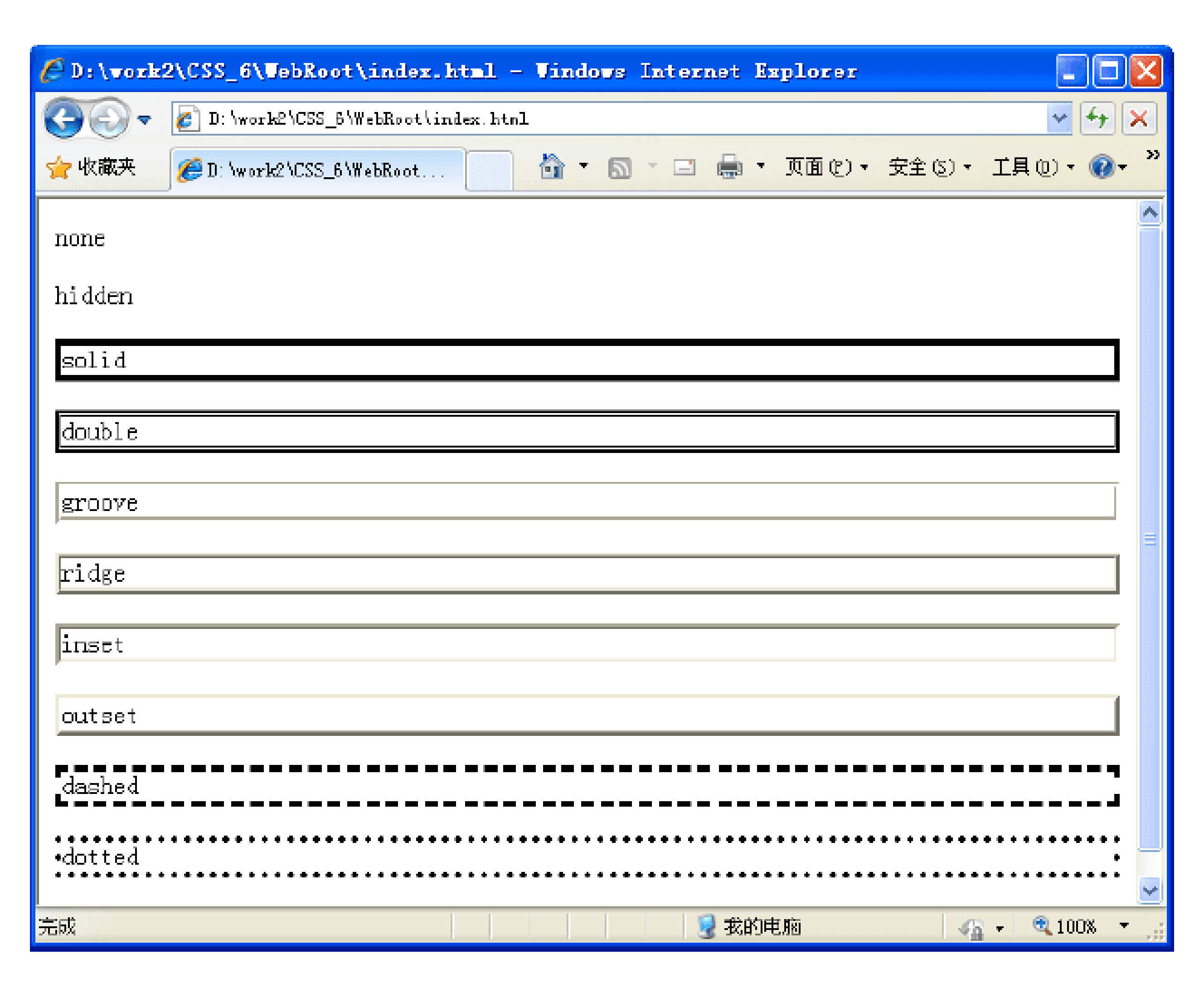


图 6-1　边框样式

6.1.2　基础知识——border-width 属性

border-width 属性为元素的所有边框设置宽度，或者单独地为各边框设置宽度。它的可选

值分别是 medium、thin 和 thick 等。具体介绍如表 6-2 所示。

表 6-2　border-width 属性

属 性 名	说　明
medium	默认值，定义中等的边框
thin	定义细的边框
thick	定义粗的边框
length	允许自定义边框的宽度
width	自定义宽度
inherit	规定应该从父元素继承边框宽度



border-width 属性只有当边框样式不是 none 时才起作用。如果边框样式是 none，边框宽度实际上会重置为 0。也不允许指定负长度值。另外所有浏览器都支持 border-width 属性，但任何版本的 Internet Explorer(包括 IE8)都不支持属性值“inherit”。

border-width 属性的语法格式：

```
border-width : medium | thin | thick | length
```

下面使用 border-width 属性的常用属性值来设置 div 元素的边框宽度，CSS 代码如下所示：

```
div.div1 {
    border-style: solid; border-width: 1px;        //设置边框宽度为 1 个像素
}
div.div2 {
    border-style: solid; border-width: 1em;        //设置边框宽度为 1em
}
div.div3 {
    border-style: solid; border-width: 1ex;        //设置边框宽度为 1ex
}
div.div4 {
    border-style: solid; border-width: thin;       //设置细的边框
}
div.div5 {
    border-style: solid; border-width: medium;     //设置中等的边框
}
div.div6 {
    border-style: solid; border-width: thick;      //设置粗的边框
}
```

设置 div 元素边框宽度。在浏览器中的执行效果如图 6-2 所示。

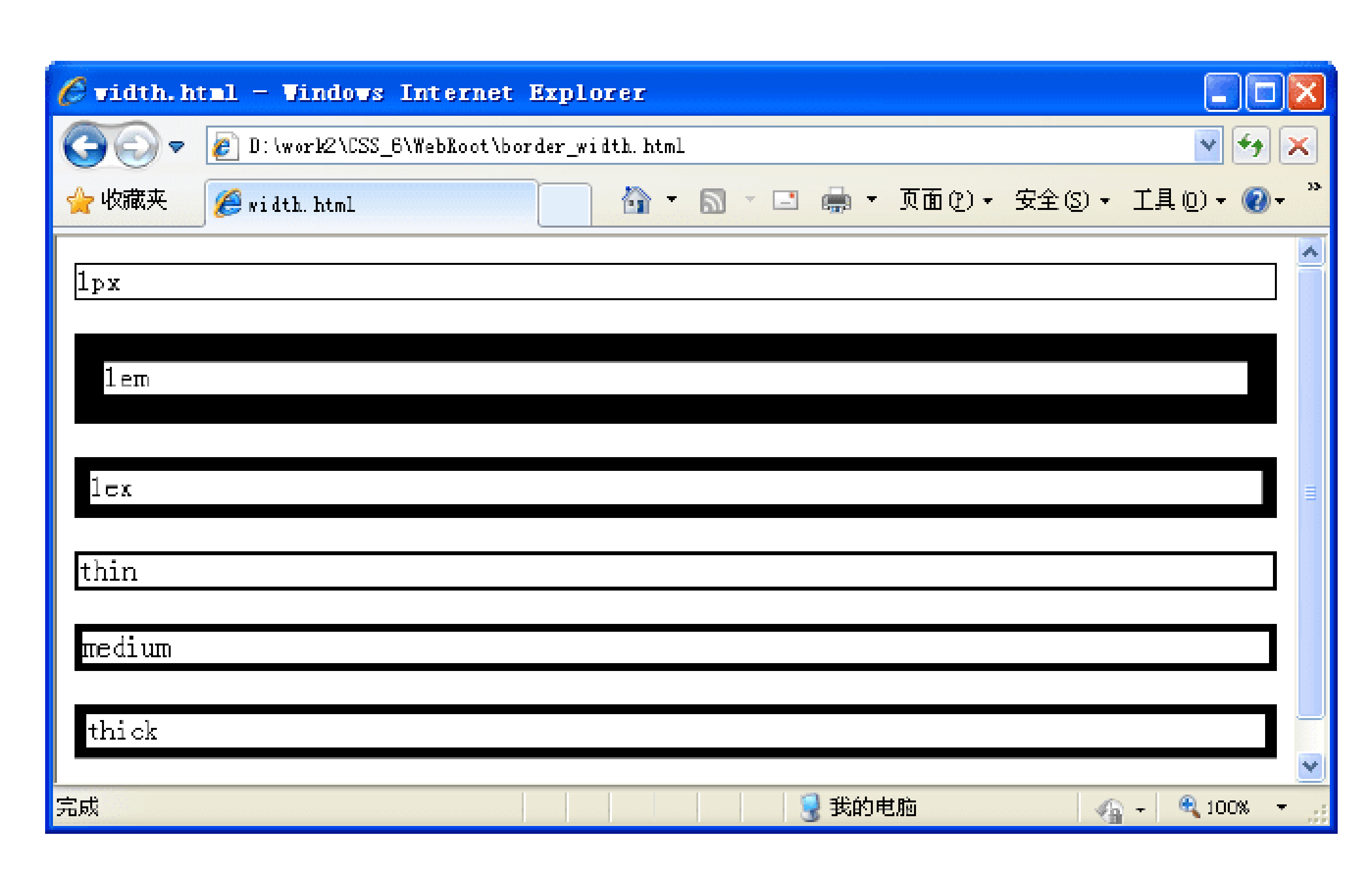


图 6-2　设置边框宽度

6.1.3　基础知识——border-color 属性

border-color 属性用于设置边框颜色。border-color 可以指定一个 color 参数来设置边框的颜色，语法格式如下：

```
border-color : color
```

下面使用 border-color 属性来设置 div 元素的边框颜色，代码如下所示：

```
.d1 {
    border-color:#FFA500;       //设置 div 边框颜色为橙色
    border-style:solid;
}
.d2 {
    border-color:#33CC00;       //设置 div 边框颜色为绿色
    border-style:solid;
}
.d3 {
    border-color:#6600FF;       //设置 div 边框颜色为蓝色
    border-style:solid;
}
.d4 {
    border-color:#FF6347;       //设置 div 边框颜色为番茄红
    border-style:solid;
}
.d5 {
    border-color:gray;          //设置 div 边框颜色为灰色
    border-style:solid;
}
```

设置 div 元素的边框颜色。在浏览器中执行效果如图 6-3 所示。

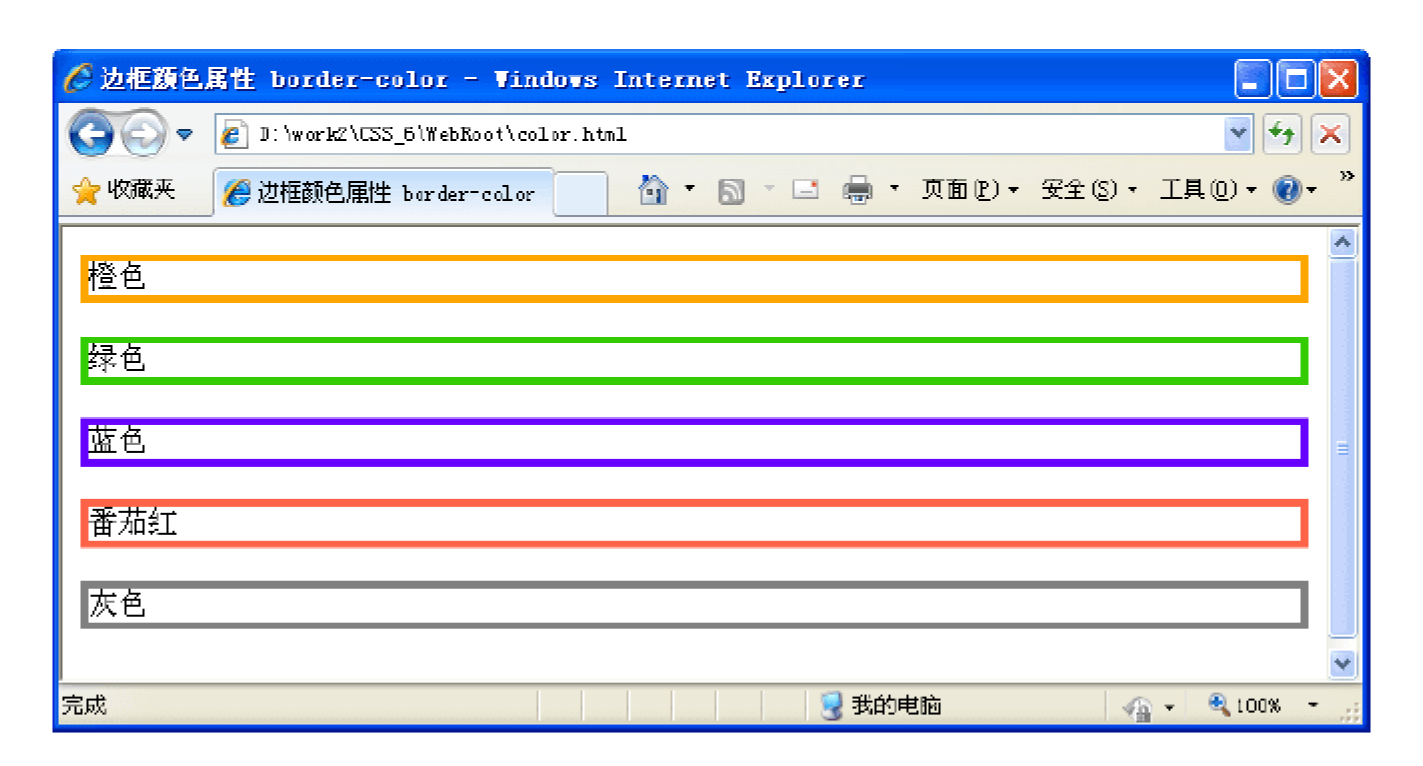


图 6-3　设置边框颜色

6.1.4　基础知识——border 属性

border 属性是一个复合属性，包含了上述 border-style、border-width 和 border-color 三种属性，可以设置页面元素的样式、宽度和颜色。语法格式如下：

```
border : border-width || border-style || border-color
```

下面使用 border 属性来设置页面 div 元素边框的样式、宽度和颜色。代码如下所示：

```
.border{
    border:thick double red;        //设置 div 元素边框为中等宽度，双线样式、红色
}
```

border 属性设置 div 元素为双线样式，中等红色的边框。在浏览器中执行效果如图 6-4 所示。



图 6-4　border 属性设置 div 元素边框

6.1.5　实例描述

本实例通过设置页面 div 元素的边框样式，最后生成一个彩色的盒子，大家一定很好奇吧，元素边框怎么会变成一个盒子呢？那就带着好奇心，一起来动手做一做这个实例吧。

6.1.6　实例应用

【例 6-1】设置一个彩色的盒子

(1) 新建一个 HTML 页面 border-example.html，在该页面中定义一个 div 元素。代码如下所示：

```
    <body>
        <div class="div1">
            <img src="role_files/happy.gif"/>
        </div>
    </body>
```

(2) 新建一个 CSS 文件 border-example.css，在该文件中设置 div 元素的边框样式，使该边框显示为一个彩色的盒子形状，代码如下所示：

```
.div1{
    height:20px;
    width:20px;
    border-color:#FF9600 #3366ff #12ad2a #f0eb7a;
    border-style:solid;
    border-width:20px;
}
```

6.1.7　运行结果

实例编写完成之后，在浏览器中运行 border_example.html 文件，可以看到一个彩色的盒子，执行效果如图 6-5 所示。



图 6-5　彩色的盒子

6.1.8　实例分析



源码解析：

本实例主要就是设置一个 div 元素的边框样式。使用 border-style 属性设置 div 元素的边框为实线边框，border-width 属性设置边框宽度为 20px，border-color 属性设置边框四边的颜色，最后一个彩色的盒子就形成了。

6.2　设置一个圆角边框

上节学习了整体设置页面元素的边框样式属性，本节将介绍一些设置局部边框的样式属性，如：border-top 属性(设置顶部边框样式)、border-right 属性(设置右边框样式)、border-bottom 属性(设置底部边框样式)和 border-left 属性(设置左边框样式)。

**视频教学：**光盘/视频/6/6.2　设置一个圆角边框　　长度：12 分钟

6.2.1　基础知识——border-top 属性

border-top 属性属于一个复合属性，用于设置顶部边框的样式，可以按顺序设置 border-top-width、border-top-style 和 border-top-color 属性。

1. border-top-width 属性

border-top-width 属性用于设置页面元素顶部边框的宽度，该属性只有当边框样式不是 none 时才起作用。如果边框样式是 none，边框宽度实际上会重置为 0。不允许指定负长度值。该属性的可选值与 border-width 属性的可选值相同，只不过 border-width 属性值是用来设置页面元素整个边框的宽度，而 border-top-width 属性值用来设置页面元素的顶部边框宽度。border-top-width 属性的基本语法格式如下所示：

```
border-top-width : medium | thin | thick | length
```

下面看一个小实例设置页面 div 元素的顶部边框的宽度，代码如下所示：

```
.border-top-width{
    width:327px;
    border:solid;
    border-top-width:10px;
}
```

设置页面 div 元素顶部边框宽度。执行效果如图 6-6 所示。

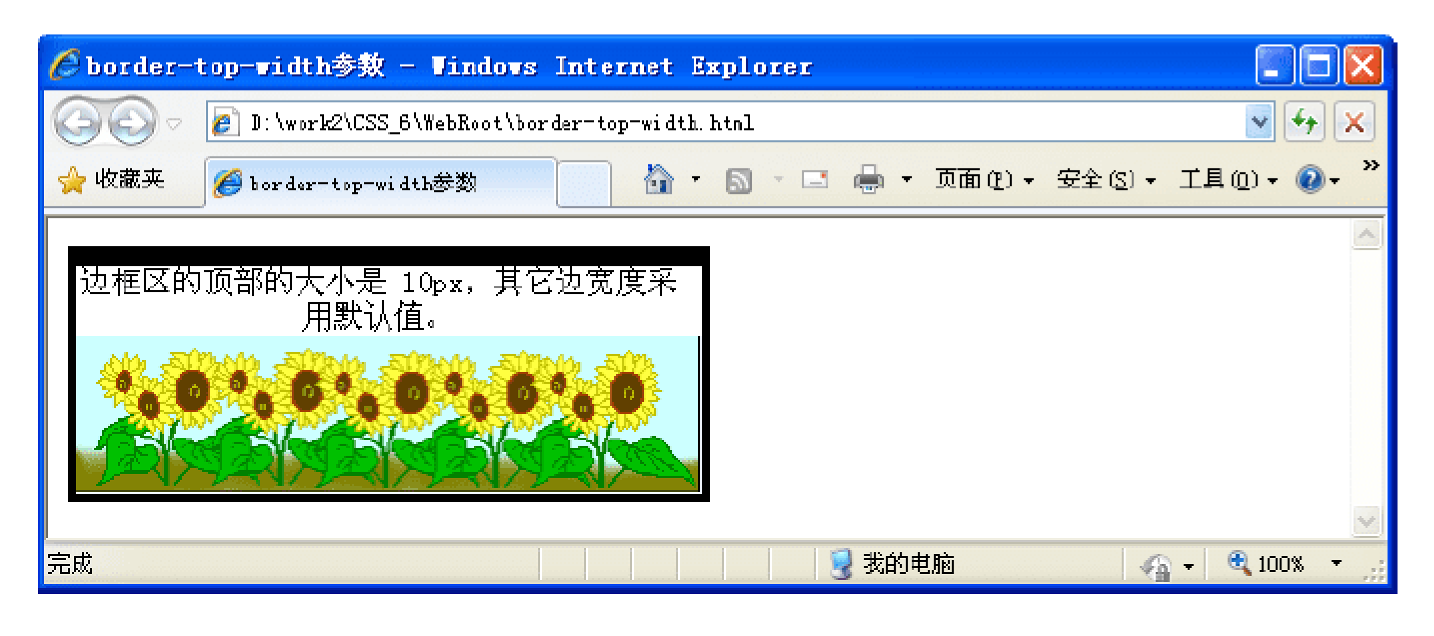


图 6-6　设置 div 元素宽度



请在 border-top-width 属性之前声明 border-style 属性。元素只有在获得边框之后，才能改变其边框的宽度。

2．border-top-style 属性

border-top-style 属性用于设置页面元素顶部边框的样式。只有当该属性值不是 none 时边框才可能出现。border-top-style 属性的可选值与 border-style 属性的可选值相同，如表 6-1 所示。

border-top-style 属性的基本语法格式如下所示：

```
border-top-style : none | hidden | dotted | dashed | solid | double | groove
| ridge | inset | outset
```

下面使用 border-top-style 属性设置页面 div 元素的顶部边框为两条蓝色直线，代码如下所示：

```
.border-top-style{
    border:blue 10px;
    border-top-style:double;
}
```

设置页面 div 元素上边框样式。执行效果如图 6-7 所示。

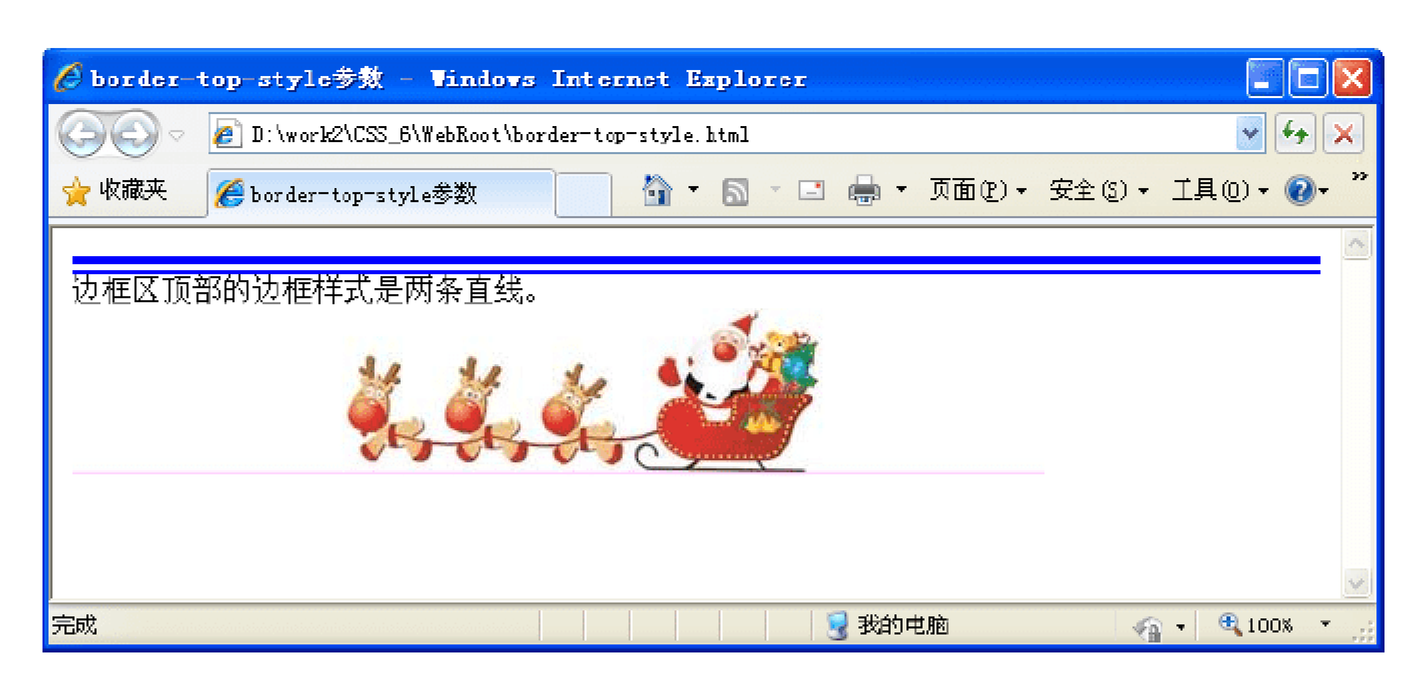


图 6-7　设置元素上边框样式

3．border-top-color 属性

border-top-color 属性用于设置页面元素顶部边框的颜色。该属性只能定义纯色的边框颜色，而且只有当边框的样式是一个非 none 或 hidden 的值时边框才可能出现。border-top-color 属性可以指定一个 color 参数值，为页面元素设置上边框的颜色，语法格式如下所示：

```
border-top-color : color
```

下面使用 border-top-color 属性设置页面 p 元素的顶部边框颜色，代码如下所示：

```
p{
    width:610px;
    border-style: solid;
    border-top-color: #ff0000;
}
```

border-top-color 属性设置页面 p 元素的顶部边框颜色的执行效果如图 6-8 所示。

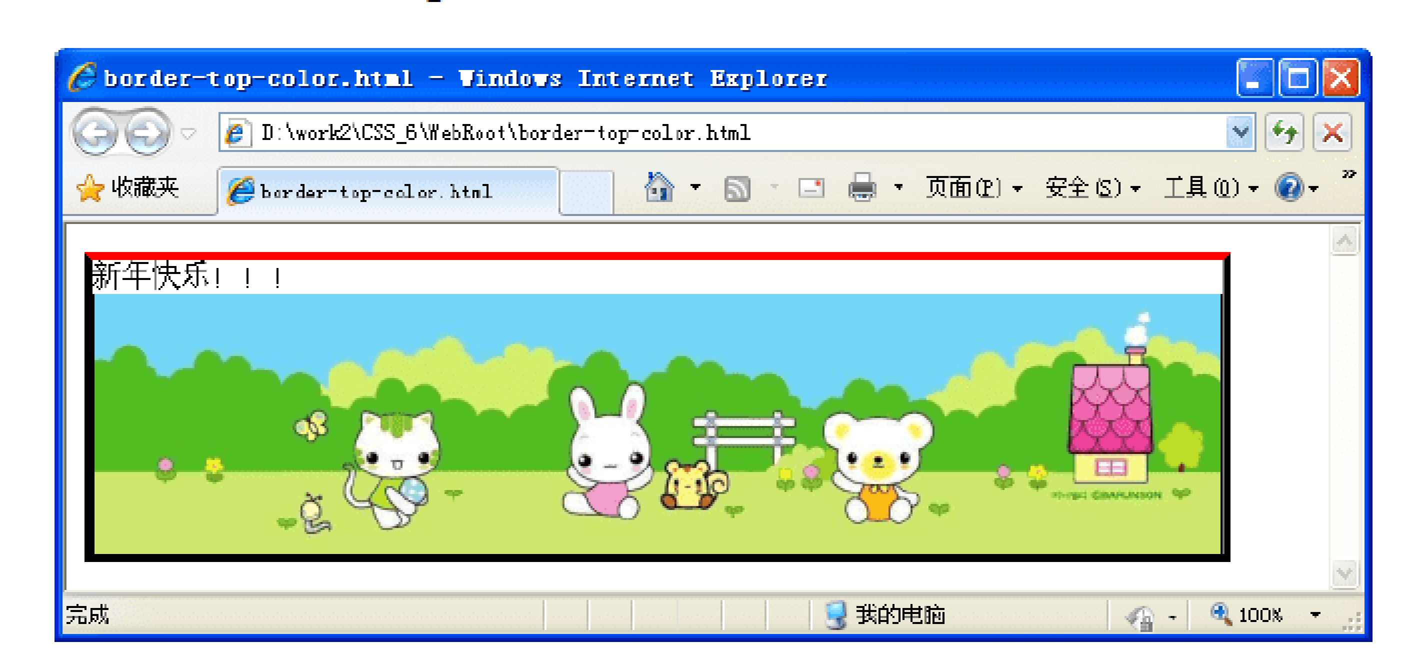


图 6-8　设置页面 p 元素的上边框颜色



一定要把 border-style 属性声明在 border-color 属性之前，因为页面元素必须在改变其颜色之前获得边框。

6.2.2　基础知识——border-right 属性

border-right 属性用于设置页面元素右边框的样式，可按顺序设置 border-right-width、border-right-style 和 border-right-color 属性。

1．border-right-width 属性

border-right-width 属性用于设置页面元素右边框的宽度。该属性也是只有当边框样式不为 none 时才起作用。如果边框样式是 none，边框宽度实际上会重置为 0。不允许指定负长度值。

border-right-width 属性的可选值也和 border-width 属性相同。只不过 border-width 属性是设置元素整体边框的宽度，而 border-right-width 属性只设置元素右边框的宽度。基本语法格式如下所示：

```
border-right-width : medium | thin | thick | length
```

下面使用 border-right-width 属性设置页面 p 元素的右边框的宽度，代码如下所示：

```
p
{
    border-style: solid;
    border-right-width: 15px
}
```

2．border-right-style 属性

border-right-style 属性用于设置页面元素右边框的样式，只有当该属性值不是 none 时边框才能出现。该属性可选值和 border-style 属性相同。它的基本语法格式如下所示：

```
border-right-style : none | hidden | dotted | dashed | solid | double | groove
| ridge | inset | outset
```

使用 border-right-style 属性设置页面 div 元素的右边框样式，代码如下所示：

```
div{
    border-style:solid;
    border-right-style:dotted;
}
```

3. border-right-color 属性

border-right-color 属性用于设置页面元素右边框的颜色，该属性也是只能定义纯色，而且只有当边框不是 none 或 hidden 时，才能起作用。它的可选值和 border-color 属性相同，基本语法格式如下所示：

```
border-right-color : color
```

使用 border-right-color 属性设置页面 div 元素的右边框的颜色，代码如下所示：

```
div{
    border-right-style:solid;
    border-right-color:green;
}
```

6.2.3　基础知识——border-bottom 属性

border-bottom 属性用于设置页面元素底部边框的样式。它是一个复合属性，可按顺序设置 border-bottom-width、border-bottom-style 和 border-bottom-color 属性。

1. border-bottom-width 属性

border-bottom-width 属性用于设置页面元素的底部边框的宽度。如果边框样式不是 none，那么该属性将才起作用，边框样式为 none，边框的实际宽度会重置为 0，不允许指定负长度值。它的基本语法格式如下所示：

```
border-bottom-width : medium | thin | thick | length
```

使用 border-bottom-width 属性设置页面 div 元素的底部边框宽度，代码如下所示：

```
div{
    border-style:solid;
    border-bottom-width:thick;      //定义粗的下边框
}
```

2. border-bottom-style 属性

border-bottom-style 属性可以设置页面元素的底部边框的样式，另外为了方便呈现 border-bottom-style 属性设置的边框样式，必须设置大于零的 border-width。border-bottom-style 属性的基本语法格式如下所示：

```
border-bottom-style : none | hidden | dotted | dashed | solid | double | groove
| ridge | inset | outset
```

使用 border-bottom-style 属性设置页面 div 元素的底部边框样式，代码如下所示：

```
div{
    border-bottom-style:solid;
    border-bottom-width:thick;
}
```

3．border-bottom-color 属性

border-bottom-color 设置元素底部边框的颜色，只能设置纯色，并且只有当边框的样式是一个非 none 或 hidden 的时候边框才能出现。该属性的使用语法如下所示：

```
border-bottom-color : color
```

下面使用 border-bottom-color 来设置 div 元素的底部边框颜色，代码如下所示：

```
.color{
    border-style:solid;
    border-bottom-color:#ff0000;
}
```

6.2.4 基础知识——border-left 属性

border-left 属性用于设置页面元素的左边框样式。它是一个复合属性，可以按顺序设置 border-left-width、border-left-style 和 border-left-color 属性。

1．border-left-width 属性

border-left-width 属性可以设置页面元素的左边框宽度，它的可选值和 border-width 属性相同，基本语法格式如下所示：

```
border-left-width : medium | thin | thick | length
```

下面使用 border-left-width 属性设置左边框宽度，代码如下所示：

```
.border-left-width{
    border-style:solid;
    border-left-width:thin;
}
```

2．border-left-style 属性

border-left-style 属性用于设置页面元素的左边框样式，它的可选值和 border-style 属性相同，基本语法格式如下所示：

```
border-left-style : none | hidden | dotted | dashed | solid | double | groove
| ridge | inset | outset
```

下面使用 border-left-style 属性设置 body 元素的左边框样式，代码如下所示：

```
body{
    border-left-style:dashed;
```

```
    border-left-width:thin;
}
```

3. border-left-color 属性

border-left-color 属性用于设置页面元素左边框的颜色，该属性也是只能定义纯色，而且只有当边框不是 none 或 hidden 时，才能起作用。它的可选值和 border-color 属性相同，基本语法格式如下所示：

```
border-left-color : color
```

下面再来使用 border-left-color 属性来设置 body 元素的左边框颜色，代码如下所示：

```
body{
    border-left-style:dashed;
    border-left-width:thin;
    border-left-color:green;
}
```

6.2.5　实例描述

通过上面的学习，相信大家对边框的各种样式设置已经掌握得差不多了，那就来做一个实例练练手吧，本节实例要设置一个圆角的边框，怎么样？很好奇吧？这个实例设置的边框与前面介绍的边框不一样了，前面都是平角的。那就让我们一起带着好奇心来做这个实例吧。

6.2.6　实例应用

【例 6-2】设置一个圆角边框

(1) 新建一个 border-layout.html 文件，在该文件中定义一些 div 元素，代码如下所示：

```
<body>
    <div class="box">
        <div class="top"></div>
        <div class="center">图书馆：1 月 30 日-小型网上书店纪念日</div>
        <div class="bot"></div>
    </div>
</body>
```

(2) 在 border-layout.html 文件所在目录下新建一个 border-layout.css 文件，用来设置该 HTML 页面元素样式生成一个圆角的边框，代码如下所示：

```
.box{
    width:500px;
}
.top{
    border-bottom:3px solid;
    border-top-color:#cc0000;
```

```
    border-bottom-color:#cc0000;
    border-left:3px dotted transparent;
    border-right:3px dotted transparent;
}
.center{
    padding:10px 20px;
    color:white;
    font-size:14px;
    background:#cc0000;
}
.bot{
    border-top:3px solid;
    border-top-color:#cc0000;
    border-bottom-color:#cccccc;
    border-left:3px dotted transparent;
    border-right:3px dotted transparent;
}
```

6.2.7 运行结果

使用 IE 浏览器打开 border-layout.html 页面，执行效果如图 6-9 所示。



图 6-9　一个圆角的边框

6.2.8 实例分析



源码解析：

本实例中生成的圆角边框是由三个 div 元素边框合成的，这三个 div 元素设置的样式名称分别是 top、center、bot，其中 top 为 div 元素设置 border-bottom、border-top-color、

border-bottom-color 等边框样式。“center”为 div 元素设置 padding、color 等属性。“bot”为 div 元素设置 border-top、border-top-color 等边框样式。这样我们的圆角边框就形成了。

6.3　使用百分比设置图像的尺寸

设置页面元素的尺寸大体上可以使用 height 和 width 属性，还可以使用 max-height、min-height 属性来设置最大高度和最小高度。本节将具体讲解这些属性的应用。

**视频教学：**光盘/视频/6/6.3　使用百分比设置图像的尺寸　　长度：7 分钟

6.3.1　基础知识——height 属性

height 属性用于设置页面元素的高度。它可以定义元素内容区的高度，在内容区外面可以增加内边距、边框和外边距，而行内非替换元素会忽略这个属性。该属性的可选值有 4 个，如表 6-3 所示。

表 6-3　height 属性值

属 性 值	说　明
auto	默认值。浏览器会计算出实际的高度
length	使用 px、cm 等单位定义高度
%	基于包含它的块级对象的百分比高度
inherit	规定应该从父元素继承 height 属性的值

height 属性的语法格式，如下：

```
height : auto | length
```

举个小例子，使用 height 属性设置一个图片元素的高度，代码如下所示：

```
img.big
{
    height: 160px    //设置 img 元素的高度
}
```

6.3.2　基础知识——max-height 属性

max-height 属性用于设置页面元素的最大高度。因此元素的实际高度值可以比指定的值小，但不能比其大，并且不允许指定负值。它的可选值有 4 个，分别是 none、length、%和 inherit。具体描述如表 6-4 所示。

表 6-4　max-height 属性值

属 性 值	说　明
none	默认值。指元素无最大高度限制
length	定义元素的最大高度值，可以由浮点数字和单位标识符组成的长度值或者百分数，不可为负数
%	定义基于包含它的块级对象的百分比最大高度
inherit	规定应该从父元素继承 max-height 属性的值

max-height 属性的语法格式如下：

```
max-height : none | length
```

下面使用 max-height 属性设置一个页面段落元素 p 的最大高度，代码如下所示：

```
.max-height{
    border:blue;
    border-style:double;
    max-height:50px;        //设置元素最大高度为 50px
}
```

6.3.3　基础知识——min-height 属性

min-height 属性用于设置页面元素的最小高度。因此元素的实际高度值可以比指定的值高，但不能比其小，且不允许指定负值。它的可选值和 max-height 属性一样有 4 个，分别是 none、length、%和 inherit。它们的使用方法也和 max-height 属性的 4 个值大致相同。在此就不再详讲。它的语法格式如下所示：

```
min-height : none | length
```

下面使用 min-height 属性再来设置一下段落元素 p 的最小高度，代码如下所示：

```
.min-height{
    border:blue;
    border-style:double;
    min-height:50px;        //设置元素最小高度为 50px
}
```

6.3.4　基础知识——width 属性

width 属性用于设置页面元素的宽度，可以定义元素内容区的宽度，在内容区外面可以增加内边距、边框和外边距。它有 4 个可选值，如表 6-5 所示。

表 6-5　width 属性值

属 性 值	说　明
auto	默认值。浏览器可计算出实际的宽度
length	使用 px、cm 等单位定义宽度
%	定义基于包含块(父元素)宽度的百分比宽度
inherit	规定应该从父元素继承 width 属性的值

width 属性的基本语法如下所示：

```
width : auto | length
```

下面我们使用 width 属性来设置 img 元素的宽度。代码如下所示：

```
<html>
    <head>
        <style type="text/css">
        img
        {
        width: 250px
        }
        </style>
    </head>
    <body>

    <img src="top.png" />
    </body>
</html>
```

图片原来为 495px，设置 width 为 250px。效果如图 6-10 所示。

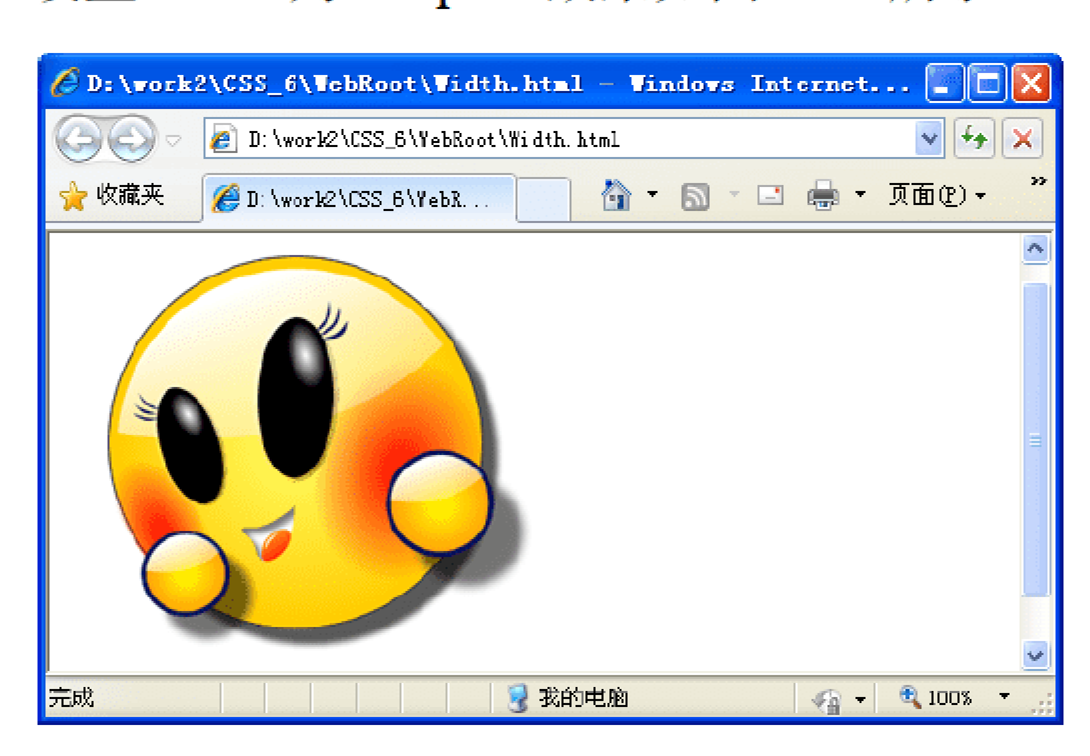


图 6-10　设置图片的 width 属性

6.3.5　基础知识——max-width 属性

max-width 属性用于设置页面元素的最大宽度，即对页面元素的宽度设置一个最高限制。因此，元素可以比指定值窄，但不能比其宽。不允许指定为负值。它的可选值有 4 个，分别是

none、length、%和 inherit。具体说明如表 6-6 所示。

表 6-6　max-width 属性值

属 性 值	说　明
none	默认值。定义对元素的最大宽度没有限制
length	定义元素的最大宽度值
%	定义基于包含它的块级对象的百分比最大宽度
inherit	规定应该从父元素继承 max-width 属性的值

max-width 属性的基本语法格式，如下所示：

```
max-width : none | length
```

下面来看看如何使用 max-width 属性来设置 div 元素的最大宽度。代码如下所示：

```
.max-width{
    border:blue;
    border-style:double;
    max-width:100px;         //设置元素最大宽度为 100px
}
```

6.3.6　基础知识——min-width 属性

min-width 属性用于设置页面元素的最小宽度，即对页面元素的宽度设置一个最小限制。因此，元素可以比指定值宽，但不能比其窄。不允许指定为负值。它有 4 个可选值，分别是 none、length、%和 inherit，如表 6-7 所示。

表 6-7　min-width 属性值

属 性 值	说　明
none	默认值。定义对元素的最小宽度没有限制
length	定义元素的最小宽度值
%	定义基于包含它的块级对象的百分比最小宽度
inherit	规定应该从父元素继承 min-width 属性的值

min-width 属性的基本语法格式，如下所示：

```
min-width : none | length
```

使用 min-width 属性来设置页面 div 元素的最小宽度，代码如下所示：

```
.min-width{
    border:blue;
    border-style:double;
    min-width:100px;         //设置元素最小宽度为 100px
}
```

6.3.7　实例描述

有时需要图片在页面中显示各种大小样式，如何设置图片大小呢？本节实例将演示使用百分比属性值来设置图像的尺寸。

6.3.8　实例应用

【例 6-3】使用百分比设置图像的尺寸

在项目 CSS_6 的 WebRoot 目录下新建一个 size.html 文件，在该文件中使用百分比属性值设置图像的尺寸，代码如下所示：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
    <head>
        <style type="text/css">
            img.normal          //自动设置图像宽高尺寸
            {
                height:auto;
                width:auto
            }
            img.big             //按 20%的比例放大图像尺寸
            {
                height: 20%;
                width:20%;
            }
            img.small           //按 10%比例缩小图像尺寸
            {
                height: 10%;
                width:10%;
            }
        </style>
    </head>
    <body>

        <img class="normal" src="images/xin.gif" />
        <br />
        <img class="big" src="images/xin.gif" />
        <br />
        <img class="small" src="images/xin.gif" />
    </body>
</html>
```

6.3.9 运行结果

在 IE 浏览器中打开 size.html 文件，可以看到图像尺寸设置效果，如图 6-11 所示。

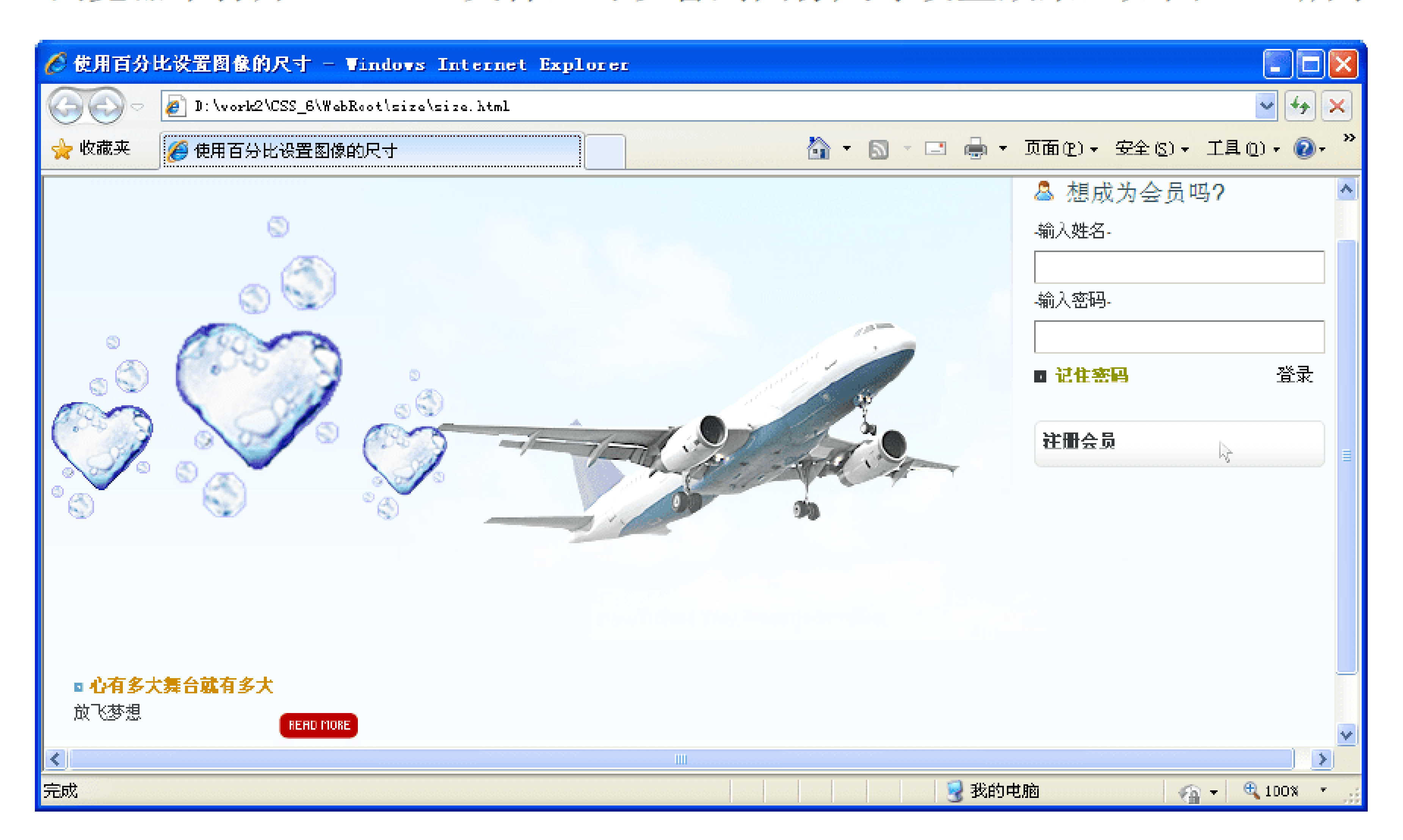


图 6-11 使用百分比设置图像的尺寸

6.3.10 实例分析



源码解析：

本实例使用 height 和 width 属性的 auto(自动调整)和百分比值来设置图像的尺寸，这样图像就可以以不同大小显示在页面中。

6.4 常见问题解答

6.4.1 CSS 边框错误



CSS 边框混乱错误。

网络课堂：http://bbs.itzcn.com/thread-12269-1-1.html

对于两个连续的边框，如果第二个边框的上边距为负值并且该边框或它的一个上级边框具有边框，则不会正确显示这些边框。这个问题怎么解决?

【解决办法】两个连续的边框？如果第二个框的上边距为负值，那么理论上它会覆盖上一个边框的一定位置。但这两个边框是独立的，如果没有在边框上使用 relative 属性，那么 Z-index

属性也不起作用。这样浏览器也不知道该显示哪一个？当然会产生冲突了。

负值的边距只能当第二个边框嵌套在第一个边框中时才会显示。或是对它们设置了 relative 才起作用。

6.4.2　CSS 的边框显示有问题



CSS 的边框显示有问题

网络课堂：http://bbs.itzcn.com/thread-12271-1-1.html

我在 CSS 中添加了这么一条语句：

```
border-bottom:solid 80%;
```

用的是傲游浏览器，但是这个边框经常不能显示，但是有时候刷新几下就可以了。请问是什么原因呢。求高手明示。

【解决办法】由于边框宽度 border-width 的属性值是不会继承的，那么就不能使用百分数来定义边框宽度 border-width。你可以参考这种模式：

```
border-bottom: 1px solid black;
```

6.5　习　　题

1. 填空题

(1) 边框的样式可以包含的值有：粗细、颜色和__________。

(2) CSS 中，div 的属性包括 margin、padding 和__________。

(3) max-height 属性用于设置页面元素的__________高度。

2. 选择题

(1) 为页面 div 元素设置底部边框颜色的属性是__________。

A. border-style　　B. border-color

C. border-bottom-color　　D. border-top-color

(2) 下面用来设置左边框样式的属性是__________。

A. border-left-style　　B. border-style

C. border-bottom-style　　D. border-right-style

(3) 使用 max-height 属性设置 div 元素最大高度为 50px 的代码正确的是__________。

A.

```
div{
    style:double;
    max-height:50px;
}
```

B.

```
div.div{
    style:double;
    max-height:50px;
}
```

C.

```
div{
    border-style:double;
    max-height:50px;
}
```

D.

```
div.div{
    border-style:double;
    max-height:50px;
}
```

(4) 边框的样式可以包含的值有__________。

A. 宽高　　B. 颜色　　C. 边距　　D. 长短

3. 上机练习

上机练习：为图片设置双边框

练习要求：首先在页面 body 元素中添加一个 img 元素，再设置 img 元素的边框，然后设置图片边框为绿色，图片与边框的间距为 5px，最后设置背景色为 red。这样看起来图片就有了双边框。执行效果如图 6-12 所示(本练习结果使用 IE 浏览器将无法完全显示，因此使用火狐浏览器)。

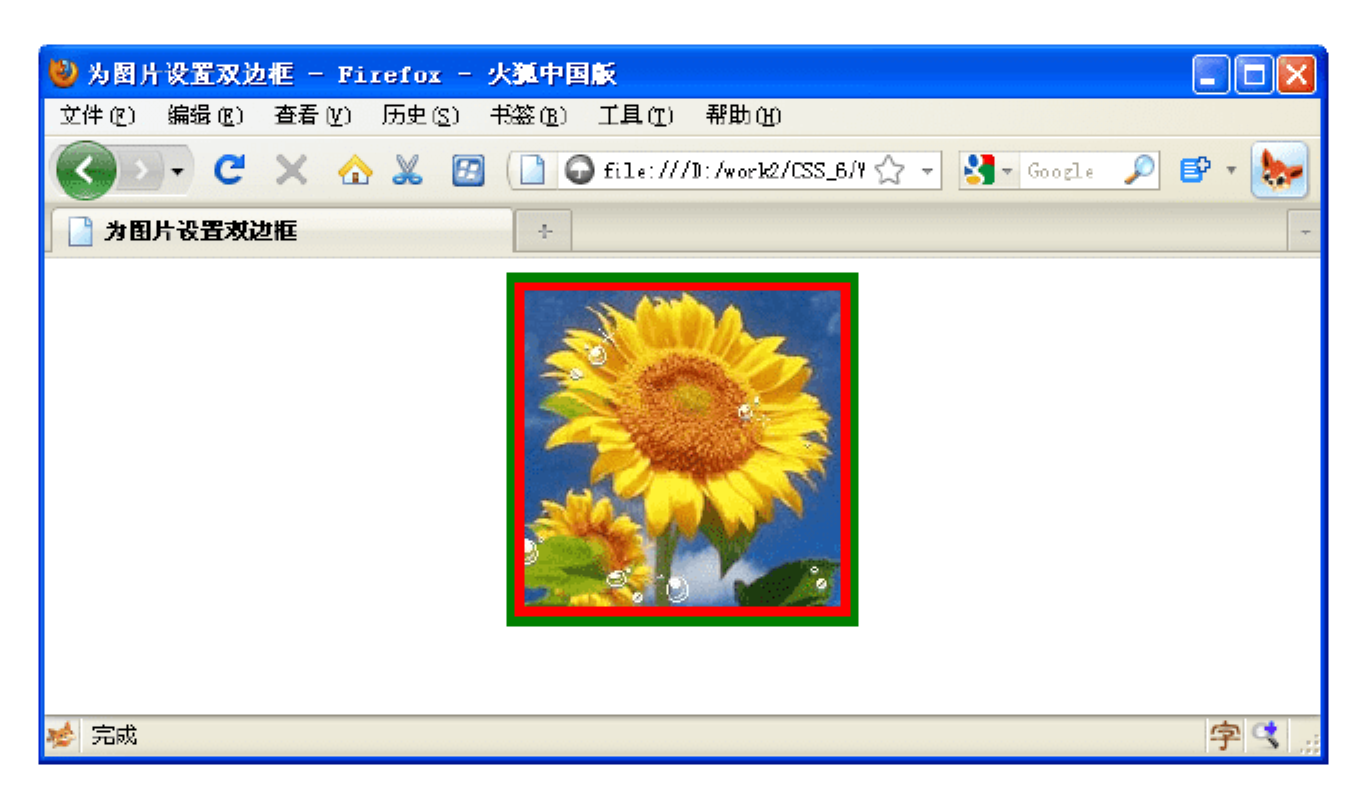
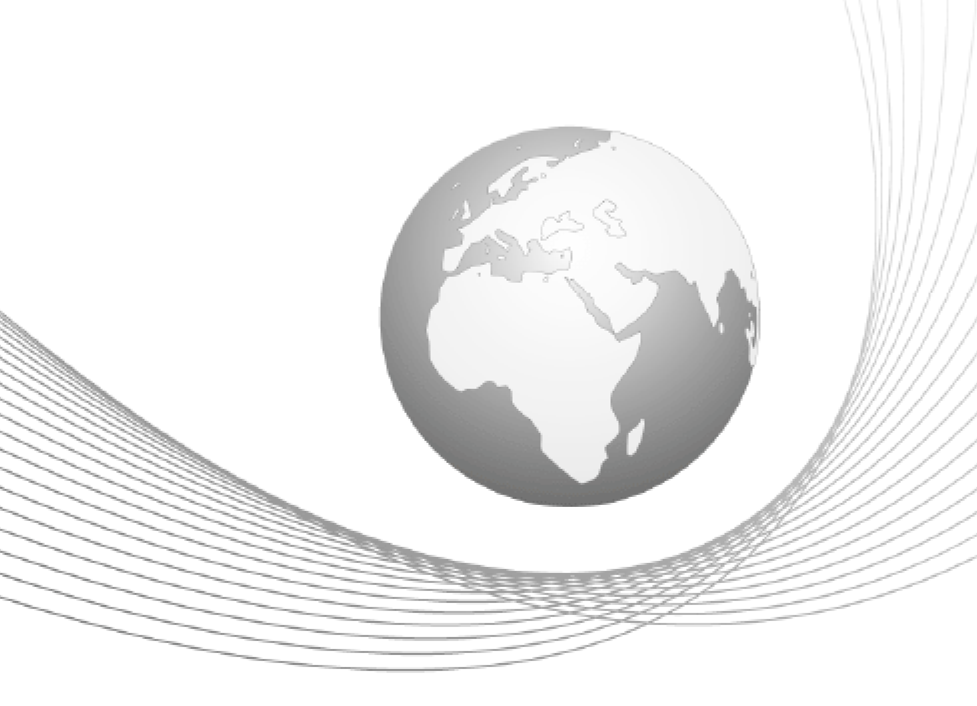


图 6-12　为图片设置双边框



第 7 章　边距与间隙属性

内容摘要：

在设计页面时，要保证页面元素出现在适当的位置，常常需要使用表格来完成。这是因为表格包含的边框能够为整个页面建立复杂的结构，而且还能使页面看起来更加美观、整洁。CSS 中的边距和间隙属性就提供了这样一种功能，能够为页面元素定义边框，并修改内容间距，从而优化文本内容的显示效果。

学习目标：

- 掌握设置元素边距的属性
- 掌握设置元素间隙的属性

7.1 边距与间隙的概念

CSS 中，所有的页面元素都包含在一个矩形框内，称为元素框。盒子模型(Box Model)规定了元素框处理元素内容、间隙、边框和外边距的方式。元素框描述了元素及其属性在页面布局中所占的空间大小，因此元素框可以影响其他元素的位置及大小。比如，页面中第一个元素框为 10px，那么下一个元素框就处于离顶部 10px 距离的位置。如果第一个元素框增加为 20px，则下一个元素框就要再下移 10px。而整个页面就是由这些大大小小，但又不会重叠的元素框构成的。

元素框最里面的部分是实际的内容，直接包围内容的是间隙。间隙呈现了元素的背景。间隙的边缘是边框。边框以外是外边距，外边距默认是透明的，因此不会遮挡其后的任何元素，示意图如图 7-1 所示。

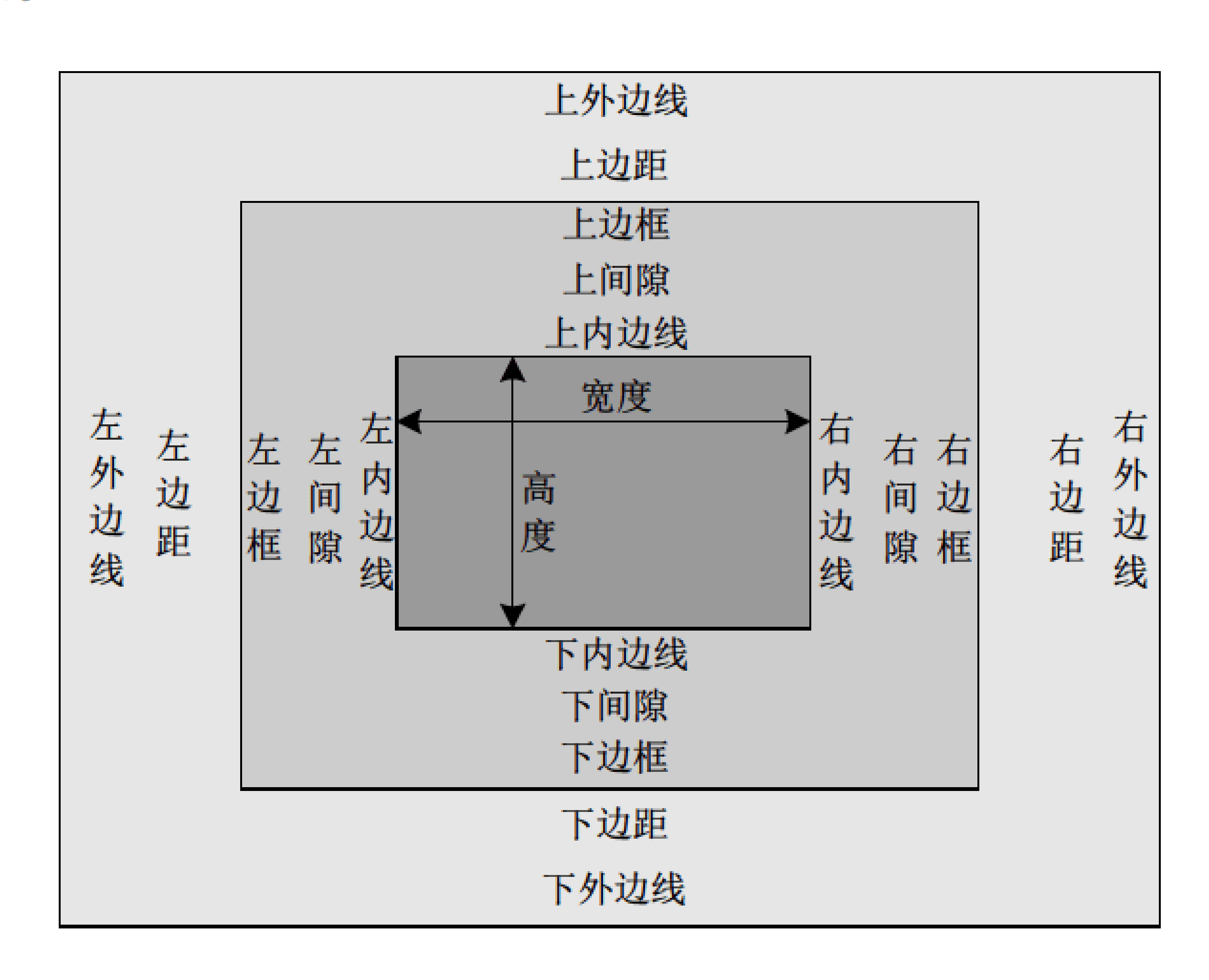


图 7-1 区块示意图

从示意图中可以清楚地看到边框与间隙所控制的区域。其中元素框就是那个最深阴影的矩形框。这时，有的读者可能会问，边距和间隙产生的作用都是定义距边界的距离，两者有什么区别呢？

其实，边距和间隙最大的区别就在于背景的显示。例如：

```
<p style="margin:20mm; background-color:yellow">
    CSS 中，所有的页面元素都包含在一个矩形框内，称为元素框。
</p>
<p style="margin:30mm; padding:5mm; background-color:yellow">
    CSS 中，所有的页面元素都包含在一个矩形框内，称为元素框。
</p>
```

在上面的代码中，第一个段落设定了边距及背景色，第二个段落设定了边距、间隙及背景色，运行该页面，效果如图 7-2 所示。

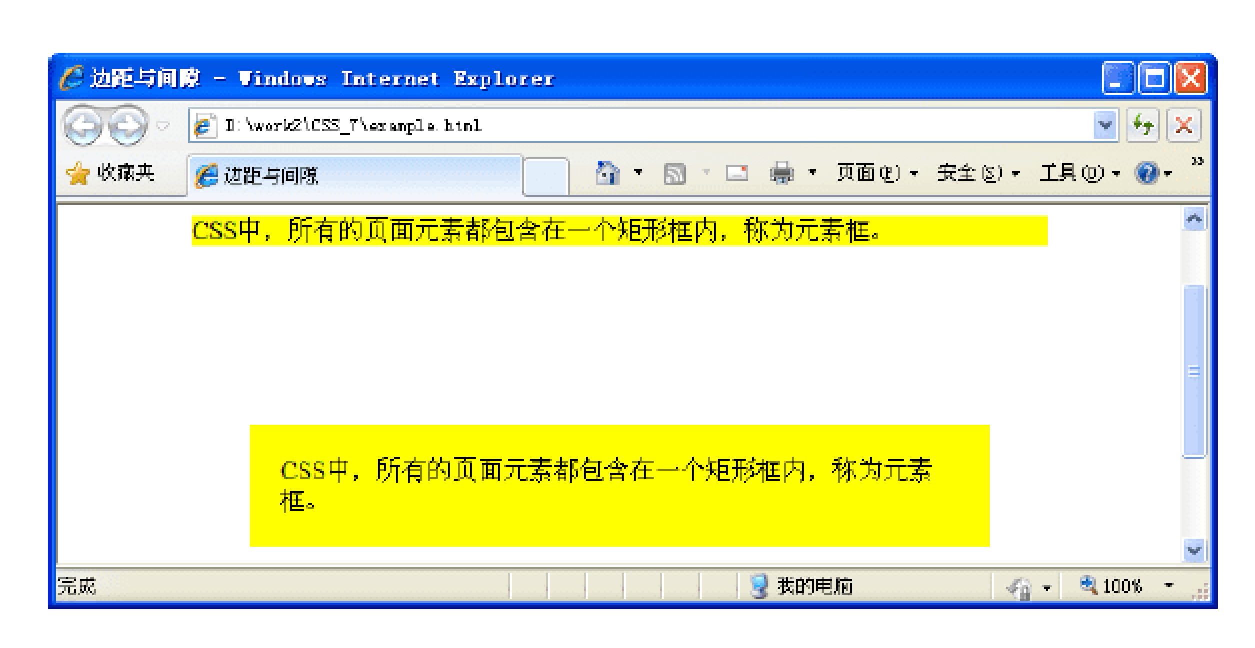


图 7-2　边距与间隙的区别

从图 7-2 中可以看出，第一个段落只设定了边距，而背景色只显示在元素框区域，这说明背景色不作用于边距。第二个段落设定了间隙，而背景色除了显示在元素框区域，也覆盖了间隙区域，这说明背景色作用于间隙。

间隙、边框和边距都是可选的，默认值是零。但是，许多元素将由用户代理样式表设置边距和间隙。可以通过将元素的 margin 和 padding 设置为零来覆盖这些浏览器样式。这可以分别进行，也可以使用通用选择器对所有元素进行设置。例如：

```
*{
    margin: 0;
    padding: 0;
}
```

7.2　水果价格表格

有时页面元素之间有一些间距，可以使页面更加简洁美观，这就是本节将要介绍的设置元素边距的属性。分别有：设置元素上外边距属性 margin-top、设置元素右外边距属性 margin-right、设置元素下外边距属性 margin-bottom 和设置元素左外边距属性 margin-left。

 **视频教学：**光盘/视频/7/7.2　水果价格表格.avi　　长度：10 分钟

7.2.1　基础知识——margin-top 属性

margin-top 属性用于设置页面元素的上外边距。该属性的可选值如表 7-1 所示。

表 7-1　margin-top 属性的取值与说明

属 性 值	说　明
auto	浏览器设置的上外边距
length	定义固定的上外边距。默认值是 0
%	定义基于父对象总高度的百分比上外边距
inherit	规定应该从父元素继承上外边距

margin 属性的基本语法格式如下所示：

```
margin-top : auto | length
```

使用 margin-top 属性来设置页面 p 元素的上外边距，代码如下所示：

```
p.topmargin {
    margin-top: 1cm
}
```

设置完毕之后，运行效果如图 7-3 所示。

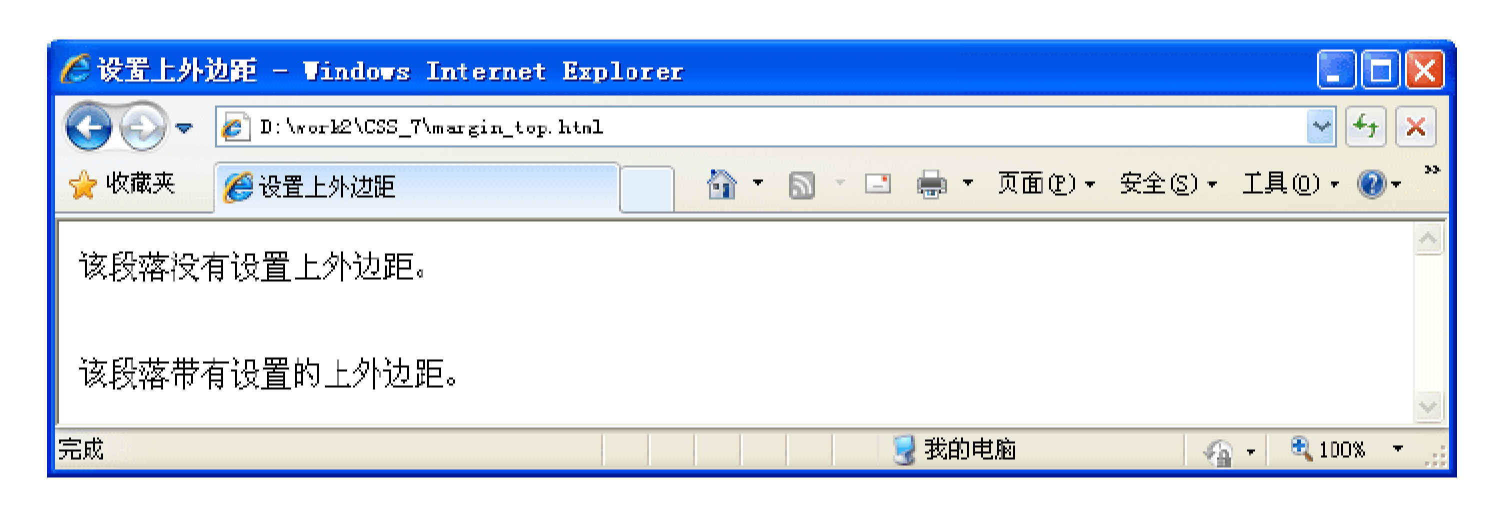


图 7-3　设置上外边距的实例运行效果

7.2.2　基础知识——margin-right 属性

margin-right 属性用于设置页面元素的右外边距。该属性的可选值如表 7-2 所示。

表 7-2　margin-right 属性的取值与说明

属 性 值	说　明
auto	浏览器设置的右外边距
length	定义固定的右外边距。默认值是 0
%	定义基于父对象总高度的百分比右外边距
inherit	规定应该从父元素继承右外边距

margin-right 属性的基本语法格式如下所示：

```
margin-right : auto | length
```

使用 margin-right 属性来设置页面 div 元素的右外边距为 12mm，代码如下所示：

```
div.rightmargin
{
    margin-right:12mm;
    background-color:yellow;
}
```

在浏览器中的运行效果如图 7-4 所示。

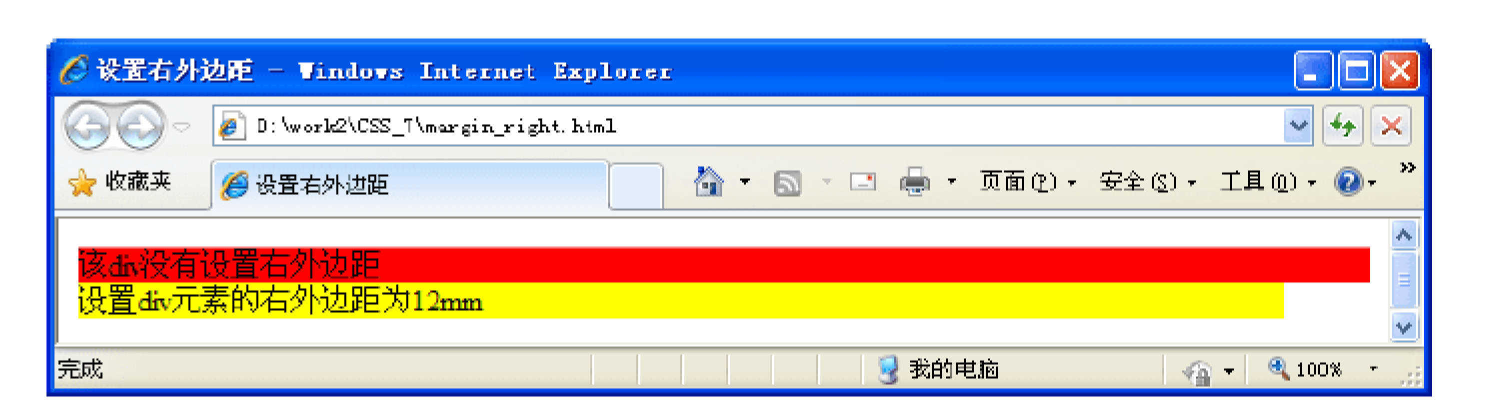


图 7-4　设置右外边距的实例运行效果

7.2.3　基础知识——margin-bottom 属性

margin-bottom 属性用于设置页面元素的下外边距，该属性的可选值如表 7-3 所示。

表 7-3　margin-bottom 属性的取值与说明

属 性 值	说　明
auto	浏览器计算下外边距
length	规定以具体单位计的下外边距值，比如像素、厘米等。默认值是 0px
%	规定基于父元素的宽度的百分比的下外边距
inherit	规定应该从父元素继承下外边距

margin-bottom 属性的基本语法格式如下所示：

```
margin-bottom : auto | length
```

使用 margin-bottom 属性设置 div 元素的下外边距为 1cm，代码如下所示：

```
.marginbottom{
    margin-bottom:1cm;
}
```

在浏览器中的运行效果如图 7-5 所示。

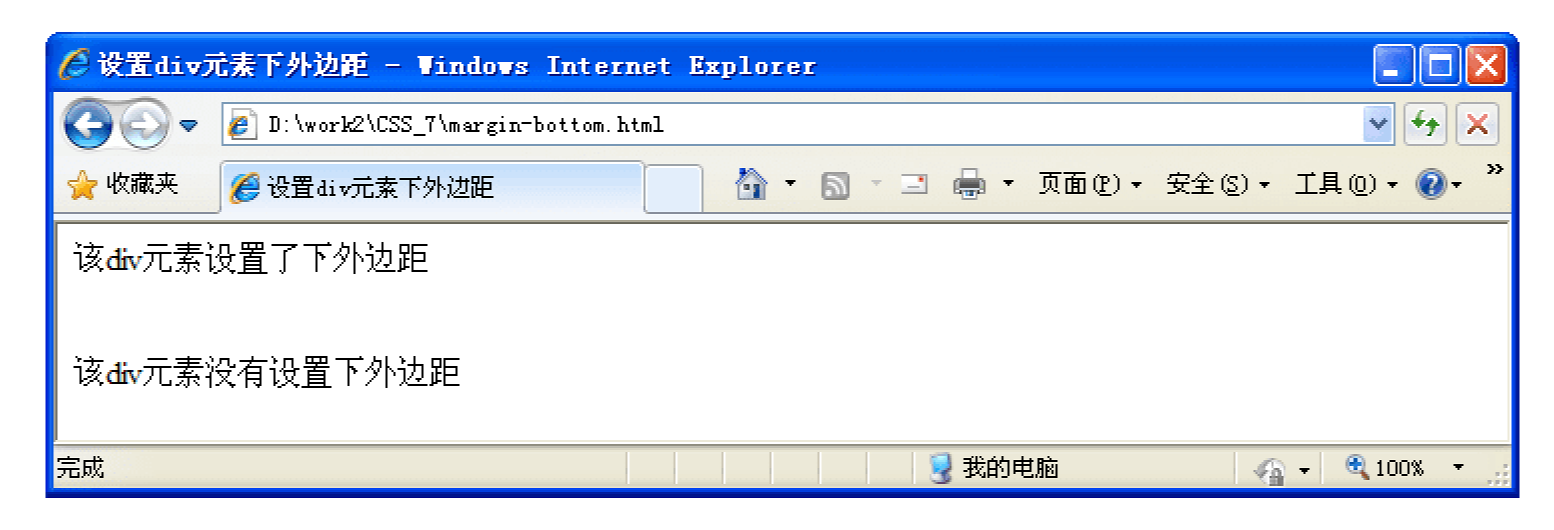


图 7-5　设置 div 元素下外边距的实例运行效果

7.2.4　基础知识——margin-left 属性

margin-left 属性用于设置元素的左外边距，它的可选属性值如表 7-4 所示。

margin-left 属性的基本语法格式如下所示：

```
margin-left : auto | length
```

表 7-4　margin-left 属性的取值与说明

属 性 值	说　明
auto	浏览器设置的左外边距
length	定义固定的左外边距。默认值是 0
%	定义基于父对象总高度的百分比左外边距
inherit	规定应该从父元素继承左外边距

下面使用百分比来设置 div 元素的左外边距，代码如下所示：

```
.marginleft{
    margin-left: 40%
}
```

在浏览器中运行的效果如图 7-6 所示。

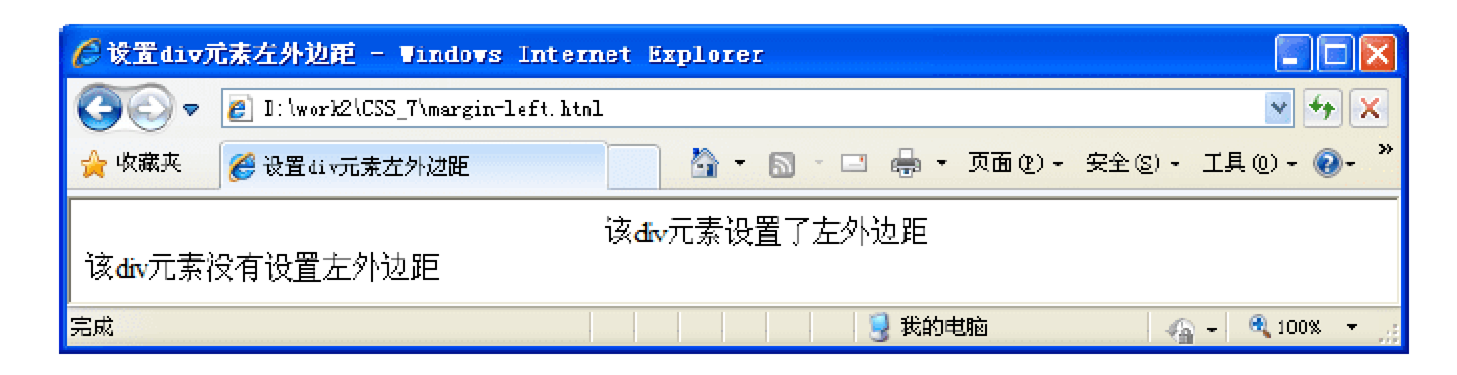


图 7-6　设置 div 元素左外边距的运行效果

7.2.5　实例描述

难得轻松一下，就到我的博客里看看，看到有人说想设置表格边距，但一直没有成功，求高手给个例子。正好刚写了本节的实例，即设置一个水果价格表格的边距，就给他发过去了，实际上没有什么特别的，就是灵活运用本节讲解的设置上下左右边距的 4 个属性，即可轻松实现。本节就把该案例的实现方法展示给大家，希望对大家有所帮助。

7.2.6　实例应用

【例 7-1】水果价格表格

(1) 新建一个 edge.html 文件，在该文件中定义一个水果价格表格，代码如下所示：

```
<table summary="Sample table">
    <caption>
        水果价格表
    </caption>
    <thead>
        <tr>
            <th abbr="team" scope="col">水果名称</th>
            <th abbr="score" scope="col">单价</th>
        </tr>
```

```
    </thead>
    <tbody>
        <tr>
            <td>苹果</td>
            <td>5.00</td>
        </tr>
        <tr>
            <td>香蕉</td>
            <td>3.50</td>
        </tr>
    </tbody>
</table>
```

(2) 在 edge.html 页面中使用<style>标签添加表格边距设置，代码如下所示：

```
<style type="text/css">
    table{
        border: 5px solid black;
        margin-left: 50px;
        margin-right: 10px;
        margin-top: 15px;
        margin-bottom: 40px;
        width: 200px;
    }
    td,th{
        background: white;
        horizontal-align: right;
    }
</style>
```

7.2.7　运行结果

在 IE 浏览器中打开 edge.html 文件，执行效果如图 7-7 所示。



图 7-7　设置表格边距的实例运行效果

7.2.8 实例分析



源码解析：

本实例综合使用 margin-top、margin-right 等属性设置表格的上外边距为 15px，下外边距为 40px。因此在执行效果中图片与表格之间有一定的间距，左外边距为 50px，右外边距为 10px。另外，还使用 border 属性设置表格的边框样式。

7.3 设置段落的整体边距

上节介绍了使用 margin-top、margin-right、margin-bottom 和 margin-left 来设置页面元素的上下左右边距。本节介绍一个使用更方便的属性 margin，它可以设置元素的上下左右边距。



视频教学：光盘/视频/7/7.3 设置段落的整体边距.avi 长度：4 分钟

7.3.1 基础知识——margin 属性

margin 可译为“外边距”或者“外补白”。它是盒子模型(box model)的基础属性。

margin 属性包括 4 个子属性，分别是 margin-top、margin-right、margin-bottom、margin-left，可以用来设置 box 的 margin area。属性 margin 可以用来同时设置 box 的四边外边距，而 margin 属性的 4 个子属性只能设置各自的外边距。margin 属性包含的四个子属性，如表 7-5 所示。

表 7-5 margin 属性的子属性及其描述

属 性 名	描 述
margin-top	设定上边距
margin-bottom	设定下边距
margin-left	设定左边距
margin-right	设定右边距

各子属性的属性值可以是一个确定的长度，也可以是一个百分比，该百分比相对于其上级元素的宽度(width)。各属性值的默认值均为 0。

margin 属性的可选值，和它的子属性的可选值相同，分别是 auto、length、%和 inherit。

margin 属性可以一次性定义上下左右的边距，语法格式如下所示：

```
margin: [margin-top] [margin-right] [margin-bottom] [margin-left]
```



四个子属性的顺序应该严格遵循，如果定义出错，则页面的布局将不会按照预计的效果显示。但是，这四个子属性的值的单位却可以不统一，例如：上边距设为 5cm，下边距设为 4mm。

如果上、下、左、右这四个边距相等，那么可以只设定一个 margin 属性值，例如：

```
margin:15mm      //设置上、下、左、右边距都为 15mm
```

7.3.2　实例描述

记得以前有个同事说段落不能设置边距。因为自己也没有特别注意过这一点，所以也不太清楚，不过通过本节实例演示得出段落元素是可以设置边距的，学习还是得亲自动手试试才能有更深入的收获呀。

7.3.3　实例应用

【例 7-2】设置段落的整体边距

(1) 新建一个 margin-example.html 文件，在该文件中定义两个段落元素，显示两首古诗，代码如下所示：

```
<p class="margin">西江月·夜行黄沙道中
                        宋 辛弃疾
            明月别枝惊鹊，清风半夜鸣蝉。
            稻花香里说丰年，听取蛙声一片。
            七八个星天外，两三点雨山前。
            旧时茅店社林边，路转溪桥忽见。
</p>
<p>《把酒问月》
        青天有月来几时，我今停杯一问之：
        人攀明月不可得，月行却与人相随？
        皎如飞镜临丹阙，绿烟灭尽清辉发？
        但见宵从海上来，宁知晓向云间没？
        白兔捣药秋复春，嫦娥孤栖与谁邻？
        今人不见古时月，今月曾经照古人。
        古人今人若流水，共看明月皆如此。
        唯愿当歌对酒时，月光长照金樽里。
</p>
```

(2) 设置 class="margin"的 p 元素的边距，代码如下所示：

```
<style type="text/css">
    .margin{
        margin: 2% 100px 0.5cm 60px;
        border: 3px solid red ;
    }
</style>
```

7.3.4 运行结果

在 IE 浏览器中打开 margin-example.html 文件，执行效果如图 7-8 所示。



图 7-8 设置段落整体边距的实例运行效果

7.3.5 实例分析



源码解析：

本实例在页面中定义了两个 p 元素，为第一个 p 元素设置了 class 属性，其值为 margin，使用 margin 属性为 class="margin"的 p 元素设置边距，使用 border 属性设置边框样式。

7.4 设置图片间隙

学习了设置元素的边距属性，那么有设置元素内部间隙的属性吗？答案是肯定的，本节就将分别讲解设置元素上间隙属性 padding-top、设置元素右间隙属性 padding-right、设置元素下间隙属性 padding-bottom 和设置元素左间隙属性 padding-left。



视频教学：光盘/视频/7/7.4 设置图片间隙.avi　　长度：6 分钟

7.4.1 基础知识——padding-top 属性

padding 属性用于设置元素上间隙的宽度，不允许指定负间隙值。该属性的可选值如表 7-6

所示。

表 7-6　padding-top 属性的取值与说明

属 性 值	说　明
length	规定以具体单位计的固定的上间隙值，比如像素、厘米等。默认值是 0px
%	定义基于父元素宽度的百分比上间隙。此值不会如预期地那样工作于所有的浏览器中
inherit	规定应该从父元素继承上间隙



对于内联元素设置的上间隙不会影响行高计算，因此，如果一个元素既有间隙又有背景，从视觉上看可能延伸到其他行，有可能还会与其他内容重叠。

padding 的基本语法格式如下所示：

```
padding-top : length
```

设置页面 div 元素的上间隙为 2cm，代码如下所示：

```
.paddingtop{
    padding-top: 2cm;        //设置 div 元素的上间隙
    border: 1px solid green;
}
```

在浏览器中运行此代码的效果如图 7-9 所示。

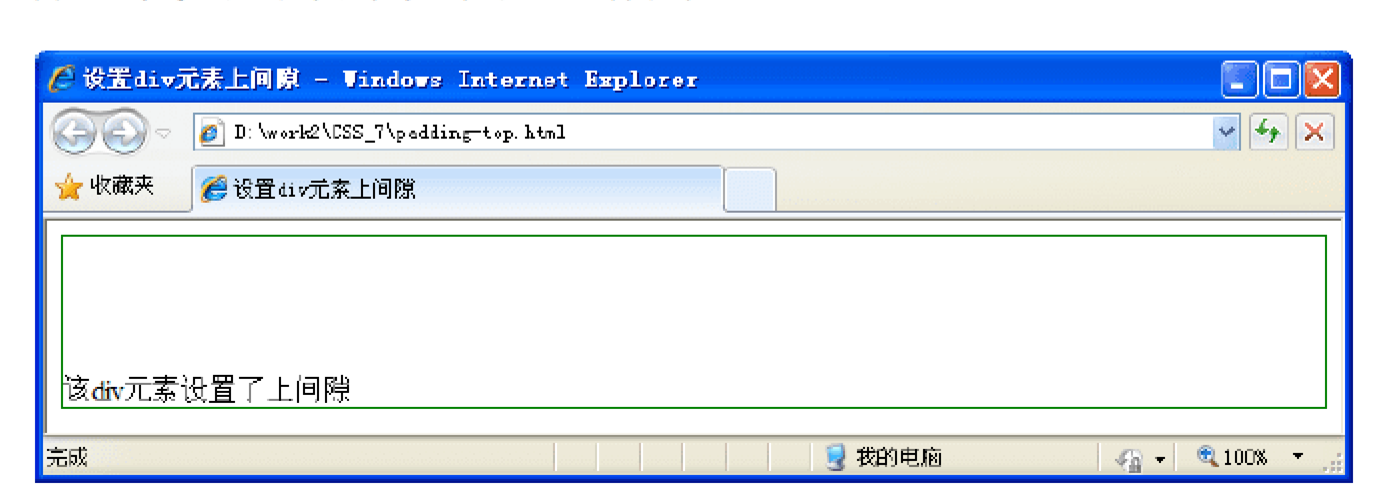


图 7-9　设置 div 元素上间隙的实例运行效果

7.4.2　基础知识——padding-right 属性

padding-right 属性用于设置元素的右间隙，不能使用负值，行内非替换元素上设置的右间隙仅在元素所生成的第一个行内框的右边出现。它的可选值如表 7-7 所示。

表 7-7　padding-right 属性的取值与说明

属 性 值	说　明
length	规定以具体单位计的固定的右间隙值，比如像素、厘米等。默认值是 0px
%	定义基于父元素宽度的百分比右间隙。此值不会如预期地那样工作于所有的浏览器中
inherit	规定应该从父元素继承右间隙

padding-right 属性的基本语法格式如下所示：

```
padding-right : length
```

设置页面 div 元素的右间隙为 10%，代码如下所示：

```
.paddingright{
    padding-right: 10%;      //设置 div 元素的右间隙
    border: 1px solid green;
}
```

在浏览器中运行此代码的效果如图 7-10 所示。

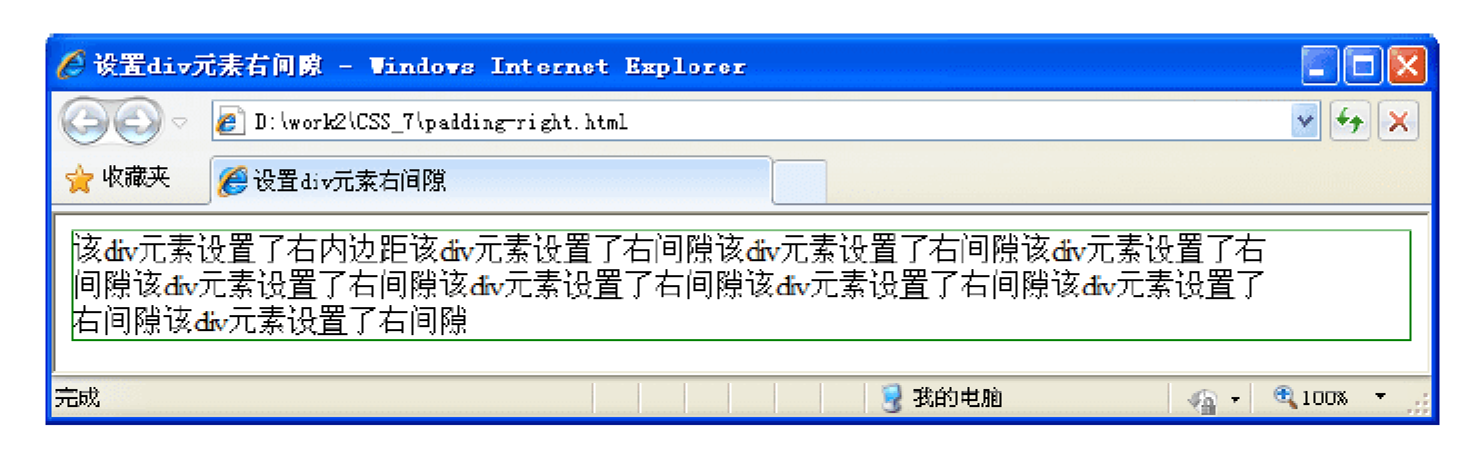


图 7-10　设置 div 元素右间隙的实例运行效果

7.4.3　基础知识——padding-bottom 属性

padding-bottom 属性用于设置元素的下间隙的宽度。也不允许指定负间隙值。它的可选值如表 7-8 所示。

表 7-8　padding-bottom 属性取值与说明

属 性 值	说　明
length	规定以具体单位计的固定的下间隙值，比如像素、厘米等。默认值是 0px
%	定义基于父元素宽度的百分比下间隙。此值不会如预期地那样工作于所有的浏览器中
inherit	规定应该从父元素继承下间隙



同上间隙一样，在行内非替换元素上设置下间隙也不会影响行高计算。因此一个元素既有间隙又有背景，从视觉上看可能延伸到其他行，有可能还会与其他内容重叠。

padding-bottom 属性的基本语法格式如下所示：

```
padding-bottom : length
```

设置 div 元素的下间隙为 20px，代码如下所示：

```
.paddingright{
    padding-right: 20px;
    border: 1px solid green;
}
```

在浏览器中运行此代码的效果如图 7-11 所示。

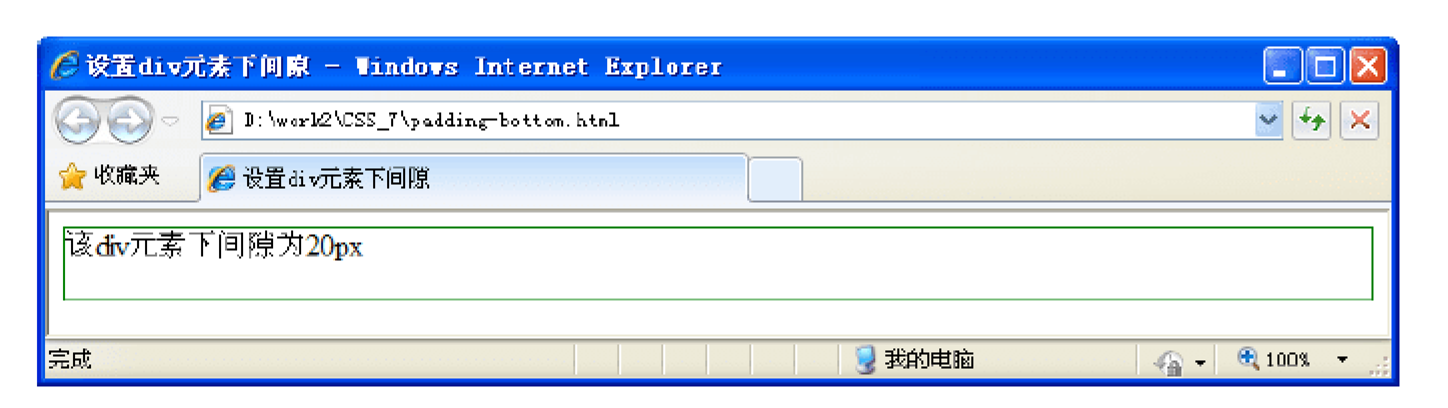


图 7-11 设置 div 元素下间隙的实例运行效果

7.4.4 基础知识——padding-left 属性

padding-left 属性用于设置元素的左间隙，行内非替换元素上设置的左间隙仅在元素所生成的第一个行内框的左边出现。该属性的可选值如表 7-9 所示。

表 7-9 padding-left 的属性的取值与说明

属 性 值	说 明
length	规定以具体单位计的固定的左间隙值，比如像素、厘米等。默认值是 0px
%	定义基于父元素宽度的百分比左间隙。此值不会如预期地那样工作于所有的浏览器中
inherit	规定应该从父元素继承左间隙

padding-left 的基本语法格式如下所示：

```
padding-left : length
```

下面使用 padding-left 属性设置 div 元素的左间隙为 30%，代码如下所示：

```
.paddingleft{
    padding-left: 30%;      //设置左间隙为 30%
    border: 1px solid green;
}
```

在浏览器中运行此代码的效果如图 7-12 所示。



图 7-12 设置 div 元素左间隙的实例运行效果

7.4.5 实例描述

前段时间在论坛中碰到了这样一个学习者，他想做个商城网站，但他做的显示商品的图片总是没有间隙，都连在一起显示，不知道该如何处理？

实际上这个问题利用本节学习的内容即可解决，下面就来看看本节的实例设置图片的间隙，使图片与文字美观地显示在页面中。

7.4.6 实例应用

【例 7-3】设置图片间隙

(1) 新建一个 img_padding.html 文件，在该文件中定义一个 div 元素，接着在 div 元素中定义一个 img 元素，代码如下所示：

```
<div id="content">
    <h1>欢迎进入绿色家园</h1>
    <p>
        <img src="images/home_img1.jpg" width="197" height="136" alt=""
title="" />
        绿色家园坐落在奉贤区金汇镇的泰日，占地面积 30000 平方米，建筑面积 50090 平方米，
        由多层住宅组成。泰日是个百年古镇，医院、银行、学校、超市，一应俱全，还有 24 小时
        医生上门护理，小区拥有两条天然河流，被绿色环绕，南面是植物园，北面是钓鱼度假村。 这
        里天蓝、地绿、水清、空新，是完美的养老胜地。
    </p>
</div>
```

(2) 新建一个 screen.css 文件，在该文件中设置 img_padding.html 文件的样式，代码如下所示：

```
h1 {
    font-size: 14px;
    font-weight: bold;
    color: #690;
}
img {
    float: right;
    padding-left: 10px;     //设置图片的左间隙
    padding-right: 10px;        //设置图片的右间隙
}

#content {
    width: 570px;
    margin: 0px 200px 5px 120px;
    padding: 10px;
    background-color: #FFF;
    border: 1px solid #DDD;
}
```

7.4.7 运行结果

代码编写完成之后，在 IE 浏览器中打开 img_padding.html 文件，执行效果如图 7-13 所示。

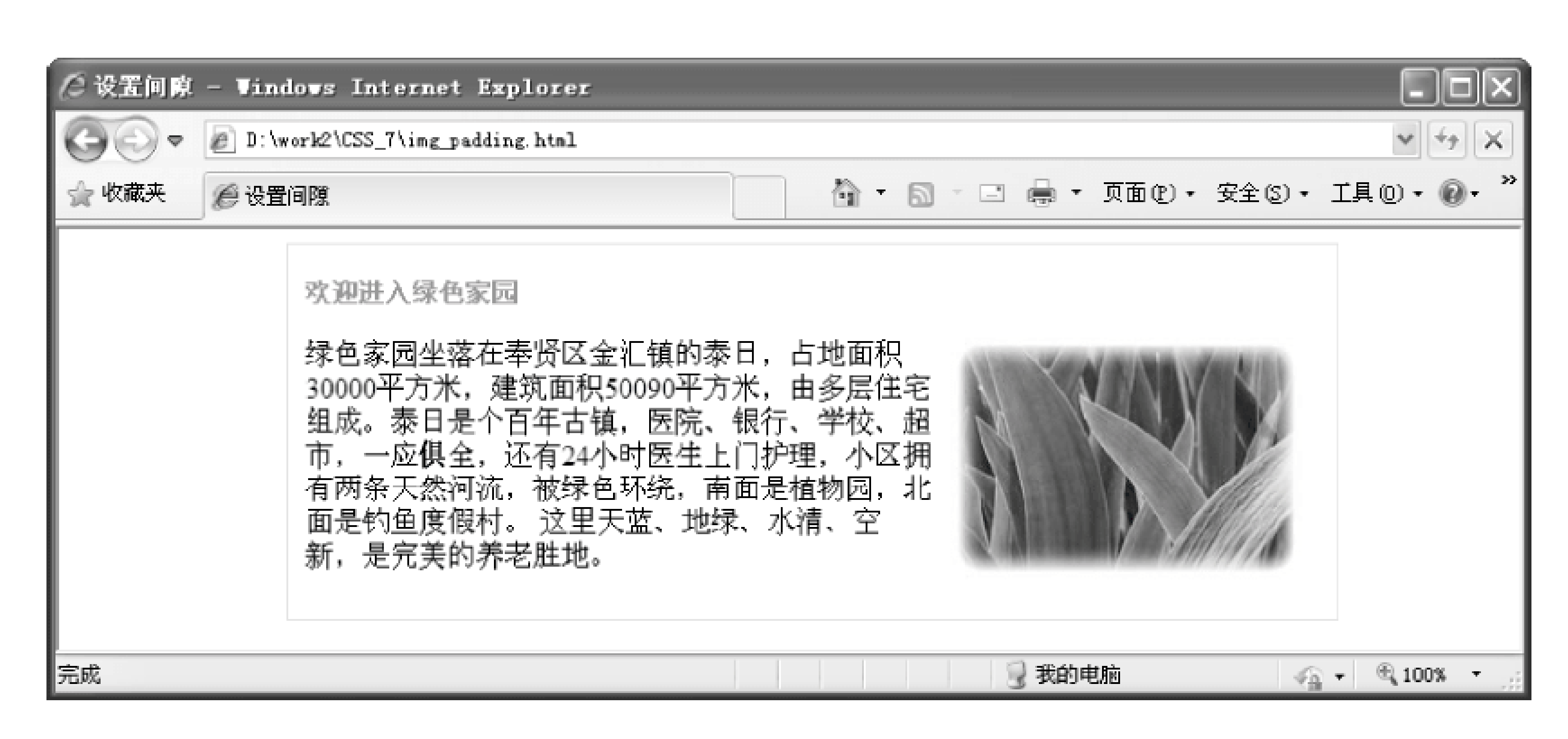


图 7-13　设置图片间隙

7.4.8　实例分析



源码解析：

本实例使用 padding-left 和 padding-right 属性设置图片的左间隙和右间隙，使图片与文字的排版整齐美观。

7.5　设置交易笔数表格的间隙

通过前面的学习，设置页面元素边距和间隙的工作都能够完成了，但是同样 CSS 也提供了一个方便设置页面间隙的属性 padding，仅使用它也可以完成设置元素上下左右间隙的工作。

 **视频教学：**光盘/视频/7/7.5　设置交易笔数表格的间隙.avi　　长度：4 分钟

7.5.1　基础知识——padding 属性

padding 属性用于设定元素框与边距之间的间隙的宽度。该属性能为元素框定义上、下、左、右间隙的宽度，也可以单独定义各方位的宽度，这就要依赖于 padding-top、padding-bottom、padding-left 和 padding-right 四个子属性，如表 7-10 所示。

表 7-10　padding 属性的子属性及其描述

子 属 性	描　述
padding-top	设定上间隙
padding-bottom	设定下间隙
padding-left	设定左间隙
padding-right	设定右间隙

padding 属性的可选值和它的子属性相同，可以是一个具体的长度，也可以是一个相对于上级元素的百分比。下面看一个简单的例子，设置一个 div 元素的上下左右间隙，代码如下所示：

```
.padding {
        padding: 1cm 10px 20% 3cm   //设置 div 元素上下左右间隙
}
```

在浏览器中运行此代码的效果如图 7-14 所示。

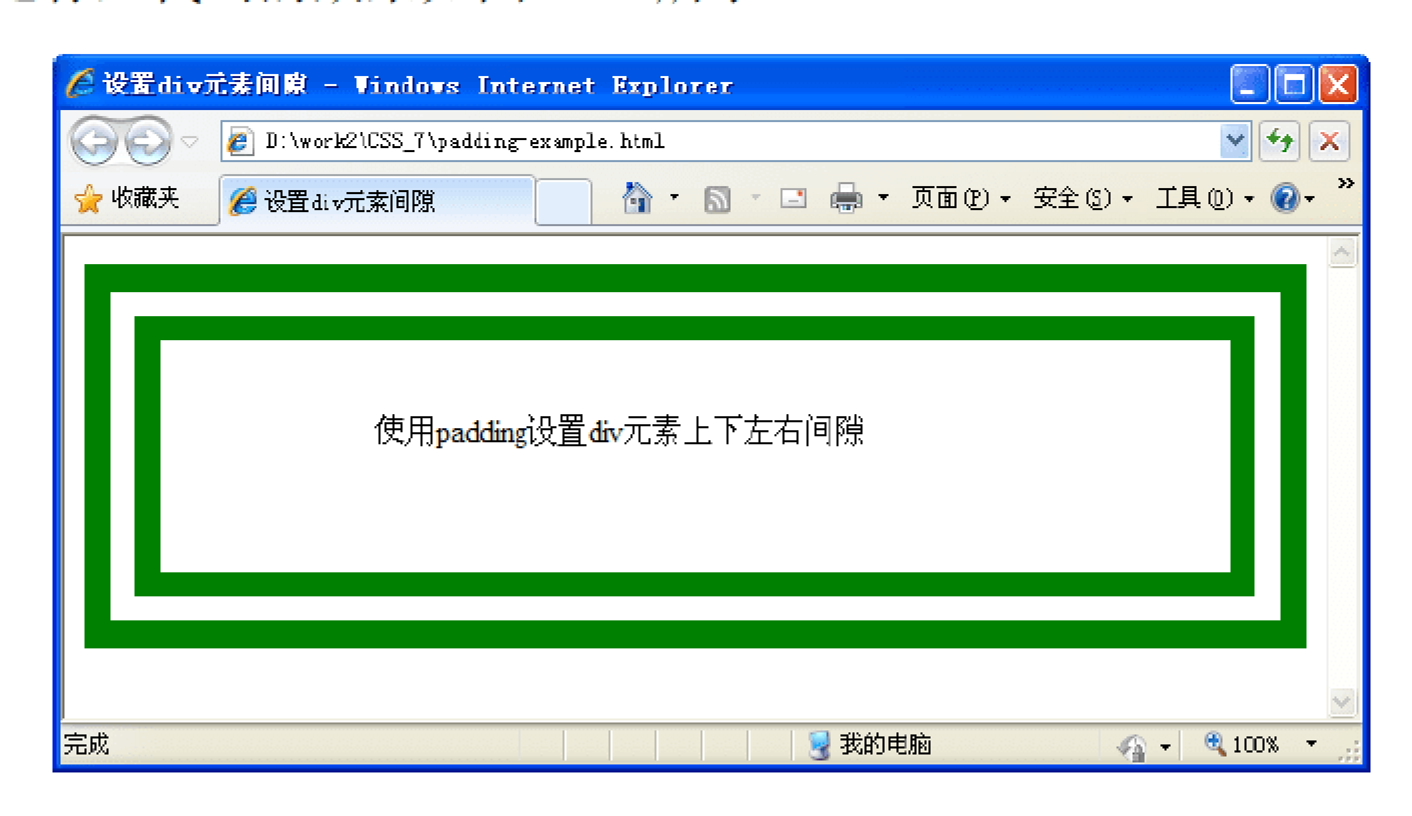


图 7-14　设置 div 元素间隙的实例运行效果

7.5.2　实例描述

在页面中显示一些统计结果数据，一般情况下采用表格的形式。但仅使用 table 元素生成的表格不太美观，为了使表格数据显示得更加整齐美观，可以通过设置表格的间隙，或表格包含的元素 tr、th 的间隙。本节实例将具体演示如何实现这些操作。

7.5.3　实例应用

【例 7-4】设置交易笔数表格的间隙

(1) 新建一个 table-padding.html 文件，在该文件中定义一个表格用于统计交易笔数数据，代码如下所示：

```
<div style="text-align:center">
    <h4>统计数据概要</h4>
    <h5><A HREF="#everyReportCountSummary">交易笔数</A></h5>
</div>
<div id="everyReportCountSummary" style="text-align:center">
    <table border=0 width="50%" height="200">
        <caption>交易笔数</caption>
        <tr>
            <th>交易状态</th><th>笔数</th>
        </tr>
```

```
        <tr>
            <td class="alt">创建笔数</td><td class="alt">123</td>
        </tr>
        <tr>
            <td>付款笔数</td><td>343</td>
        </tr>
        <tr>
            <td class="alt">发货笔数</td><td class="alt">122</td>
        </tr>
        <tr>
            <td>确认收货笔数</td><td>133</td>
        </tr>
        <tr>
            <td class="alt">申请退款笔数</td><td class="alt">50</td>
        </tr>
        <tr>
            <td>退款成功笔数</td><td>40</td>
        </tr>
    </table>
</div>
```

(2) 使用<style>标签在 table-padding.html 文件中添加表格间隙样式，代码如下所示：

```
<style type="text/css">
    body {
        font: normal 11px auto "Trebuchet MS", Verdana, Arial, Helvetica,
sans-serif;
        color: #4f6b72;
    }
    a {
        overflow:visible;
        font-family:Tahoma,Helvetica,Arial,"宋体",sans-serif;
        color:#0047D8;
    }
    a:hover{color: #9997D8;}
        #mytable {
        width: 700px;
        padding: 0;              //设置链接间隙
        margin: 0;
    }
    caption {
        padding: 0 0 5px 0;     //设置表格 caption 元素的间隙
        width: 700px;
        font:  18px Bold "Trebuchet MS", Verdana, Arial, Helvetica, sans-serif;
        text-align: center;
    }
    th {
        font: bold 11px "Trebuchet MS", Verdana, Arial, Helvetica, sans-serif;
        color: #4f6b72;
        border-right: 1px solid #C1DAD7;
```

```
        border-bottom: 1px solid #C1DAD7;
        border-top: 1px solid #C1DAD7;
        letter-spacing: 2px;
        text-transform: uppercase;
        text-align: left;
        padding: 6px 6px 6px 12px;          //设置表格 th 元素的间隙
        background: #CAE8EA url(images/bg_header.jpg) no-repeat;
    }
    th.nobg {
        border-top: 0;
        border-left: 0;
        border-right: 1px solid #C1DAD7;
        background: none;
    }
    td {
        border-right: 1px solid #C1DAD7;
        border-bottom: 1px solid #C1DAD7;
        background: #fff;
        font-size:11px;
        padding: 6px 6px 6px 12px;          //设置表格单元格的间隙
        color: #4f6b72;
    }
    td.alt {
        background: #F5FAFA;
        color: #797268;
    }
</style>
```

7.5.4 运行结果

在 IE 浏览器中打开 table-padding.html 文件，执行效果如图 7-15 所示。



图 7-15 设置表格间隙的实例运行效果

7.5.5　实例分析



源码解析：

本实例在页面 table-padding.html 中定义了一个统计交易笔数数据的表格，使用 padding 属性设置表格的 caption 元素、th 元素和 td 元素的间隙，使表格中的数据显示得更加整齐美观。

7.6　常见问题解答

7.6.1　消除两个 div 之间的间隙问题



消除两个 div 之间的间隙问题。

网络课堂：http://bbs.itzcn.com/thread-12391-1-1.html

我想这样布局网页：两个<div>，每个 div 中含有一个
...。但是我做好后，发现无论如何设置这两个 div 的样式，两个块之间总有间隔。我现在想消除这个间隔，该如何处理？

我的完整代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>布局</title>
<style>
#a {
 margin:0;
 width:700px;
 background-color: #F00
}
#b {
 margin:0;
 width:700px;
 background-color: #0F0
}
</style>
</head>

<body>

<div id="a">
```

```
  <ul>
    <li>a1</li>
    <li>a2</li>
  </ul>
</div>

<div id="b">
  <ul>
    <li>b1</li>
    <li>b2</li>
  </ul>
</div>

</body>
</html>
```

【解决办法】我看过你的代码了，其他的都没有问题，你只要在样式表里加上以下代码就可以了。因为表单默认是有间距的，要设置它的间距为 0。

```
ul{padding:0px; margin:0px;}
```

7.6.2 margin 边距问题



margin 边距问题。

网络课堂：http://bbs.itzcn.com/thread-12392-1-1.html

请问 CSS 的 margin 边距问题？在火狐浏览器里都有上下边距，为什么在 IE 浏览器中没有上下边距了？我设置的这个不管用，代码如下所示：

```
<style>
    body{
        margin:0px;
    }
    div{
        border:1px solid red;
        line-height:16px;
    }
    p{
        height:20px;
        border:1px solid green;
        background:yellow;
        margin-top:100px;
        margin-bottom:100px;
    }
</style>
<div>
    <p>22</p>
</div>
```

请高手解答。

【解决办法】为 p 元素添加样式“margin:0; padding:0”。这样火狐浏览器中的上下边距就没有了，这是因为火狐浏览器相比 IE 浏览器要严谨一些。

7.6.3　margin-top 属性遇到点小问题



用 CSS 的 margin-top 属性遇到点小问题。

网络课堂：http://bbs.itzcn.com/thread-12393-1-1.html

在一个单元格中放入一个文本框和一个搜索按钮，但是它们不能水平对齐，我想用 margin-top 属性可以解决这个问题。

代码如下：

```
<td width="230" height="51" align="left" valign="middle">
    <input name="SearchText" type="text"
        style="border: 1px solid #CCCCCC; margin-left:12px"
         size="18" maxlength="100"/>
        <img src="images/search1.jpg"
            style="margin-top:10px" width="40" height="25" />
</td>
```

可现在是文本框和按钮一起向下移动 10px，这个问题怎么解决？

【解决办法】只需修改为下面代码即可：

```
<img src="images/search1.jpg"
    style="position:relative;top:10px" width="40" height="25" />
```

7.7　习　　题

1. 填空题

(1) padding 属性用于设置元素的间隙，它的子属性有 padding-top、padding-right、__________和 padding-left。

(2) margin 属性可以设置页面元素的__________。

(3) 下面代码使用 margin 属性设置 p 元素的边距，代码如下所示：

```
.margin{
    margin: 10% 100px 2cm 60px;
    border: 3px solid red ;
}
```

运行之后，p 元素的右边距为__________。

2. 选择题

(1) 下列 CSS 属性中能够设置盒模型的内补丁为 10px、20px、30px、40px(顺时针方

向)的是___________。

A. padding:10px 20px 30px 40px　　B. padding:10px 1px
C. padding:5px 20px 10px　　D. padding:10px

(2) 下列属性中能够设置盒模型的左侧外补丁的是___________。

A. margin　　B. indent　　C. margin-left　　D. text-indent

(3) 定义盒模型外补丁的时候是否可以使用负值，___________。

A. 是　　B. 否

(4) 下面关于 CSS 中 BOX 的 padding 属性不包括的属性值是___________。

A. 填充　　B. 上边距　　C. 底间隙　　D. 左边距

(5) CSS 中，盒模型的属性包括___________。

A. font　　B. margin　　C. text　　D. visible

3. 上机练习

上机练习：为图片制作一个金相框

练习要求：首先需要在页面中添加一个 div 元素，然后在 div 元素中添加一个图片元素，最后通过设置 img 元素的边距来形成一个相框。执行效果如图 7-16 所示。



图 7-16　制作一个金相框



第 8 章　颜色与背景属性

内容摘要：

任何网页都离不开背景的修饰和丰富的色彩。没有颜色的网页空洞而没有生机，就像一辆汽车，只有 4 个轮子和方向盘一样，这就要求一个优秀的网页设计者不仅要能够合理地安排页面布局，还要具有一定的视觉色彩和颜色的搭配能力，这样才能够使得网页更加精美，也更具有表现力，给浏览者带来亲切的感觉。

现在依赖 CSS 的强大功能，能够为页面增加更多的色彩和背景选择，并且能够统一页面元素的色彩配置，CSS 就像美丽的汽车外壳，经过汽车修理师的精心修改，能够将一个仅仅只有 4 个轮子的汽车做成万人瞩目的“法拉利”，本章将会介绍 CSS 的颜色和背景属性。

学习目标：

- 设置对象的文本颜色
- 设置对象的背景图片
- 设置背景图片的位置
- 设置背景图片的平铺方式
- 设置整个区域的背景颜色
- 设置整个区域的背景图片

8.1 后台管理系统

为了使页面布局更加合理，必须精心安排各页面元素的位置、定义页面的颜色搭配以及字体的大小、格式等。颜色包括字体以及背景的颜色，CSS 中为开发人员预设了 16 种颜色及其衍生色，这 16 种颜色是 CSS 规范推荐的，而且一些主流的浏览器都能够识别它们，本节通过后台管理系统的页面布局进行讲解。

**视频教学：**光盘/视频/8/8.1 后台管理系统.avi 长度：4 分钟

8.1.1 基础知识——background-color 属性

background-color 属性设置或检索对象的背景颜色。background-color 属性接收任何有效的颜色值，而对于没有设定背景色的标记，默认背景色为透明。下面通过几段代码来讲解如何在实例中应用 background-color 属性，详细代码如下所示：

```
<p style="background-color:PaleGreen; color:Blue">
  该属性填充的背景是指元素框及间隙所包含的区域，并位于前景的后面。
</p>
```

如果在主体(<body>标记)中定义了背景色，则该背景色将应用于整个页面中，标记代码如下：

```
<body style="background-color:PaleGreen; color:Blue">
    <p>
        该属性填充的背景是指元素框及间隙所包含的区域，并位于前景的后面。
    </p>
</body>
```

同样，background-color 属性可以在 div 元素中设置其背景颜色，还可以设置表格等很多容器的背景颜色。

8.1.2 实例描述

下面将以后台管理系统为例对页面优化的操作方法进行详细讲解。在本案例中，首先需要将列表页面中的所有字体颜色设置为黑色，将特殊数据以红色高亮显示。将列表表格背景颜色设置为#FFFFFF，当执行链接操作、未链接、鼠标经过、激活操作以及已访问后等操作时的颜色将发生改变。

8.1.3 实例应用

【例 8-1】后台管理系统

首先在页面列表中设置所有字体颜色为黑色，特殊字符则用红色高亮显示。在这里只需要设置 color 属性即可，详细代码如下所示：

```
    <tr style="color:red; background-color:#FFFFFF">
        <td height="18" bgcolor="#FFFFFF"><div align="center" class="STYLE1">
            <input name="checkbox" type="checkbox" class="STYLE2" value="checkbox" />
        </div></td>
            <td height="18"
bgcolor="#FFFFFF" class="STYLE2"><div align="center" class="STYLE2 STYLE1">
A0012</div></td>
            <td height="18"
bgcolor="#FFFFFF"><div align="center" class="STYLE2 STYLE1">192.168.0.124
</div></td>
            <td height="18"
bgcolor="#FFFFFF"><div align="center" class="STYLE2 STYLE1">wtz_fh</div></td>
            <td height="18"
bgcolor="#FFFFFF"><div align="center" class="STYLE2 STYLE1">DBserver</div>
</td>
            <td height="18" bgcolor="#FFFFFF"><div align="center" ><a
href="#">服务器进程配置</a></div></td>
            <td height="18" bgcolor="#FFFFFF"><div align="center"><img
src="images/037.gif" width="9" height="9" /><span class="STYLE1"> [</span><a
href="#">编辑</a><span class="STYLE1">]</span></div></td>
        <td height="18" bgcolor="#FFFFFF"><div align="center"><span class=
"STYLE2"><img src="images/010.gif" width="9" height="9" /> </span><span class=
"STYLE1">[</span><a href="#">删除</a><span class="STYLE1">]</span></div></td>
    </tr>
```

在上述代码中，通过设置 tr 标记的 color 属性将特殊数据的信息高亮显示，在这里设置行的字体颜色为红色，设置表格的背景颜色为白色。下面代码中则是设置启动超链接时的字体颜色变化，详细代码如下所示：

```
<style type="text/css">
    a:link{/*默认时超链接颜色*/
        color:#06482a;
        text-decoration:none
    }
    a:visited/*访问后的超链接颜色*/
    {
        color:RGB(153,102,153);
        text-decoration:none
    }
    a:hover/*鼠标移上去时颜色*/
    {
        color:#0099CC;
        text-decoration:underline
    }
    a:active/*鼠标单击不松时颜色*/
    {
```

```
        color:#CC6699;
        text-decoration:underline
    }
</style>
```

在代码中通过 color 属性，设置默认超链接颜色为#06482a，当用户将鼠标移到超链接上时将颜色设置为#0099CC，当单击超链接时颜色为#CC6699，最后则是访问后超链接颜色为 RGB(153,102,153)。

8.1.4 运行结果

接下来是保存修改页面，在浏览器中运行 main.html 页面，最终效果如图 8-1 所示。

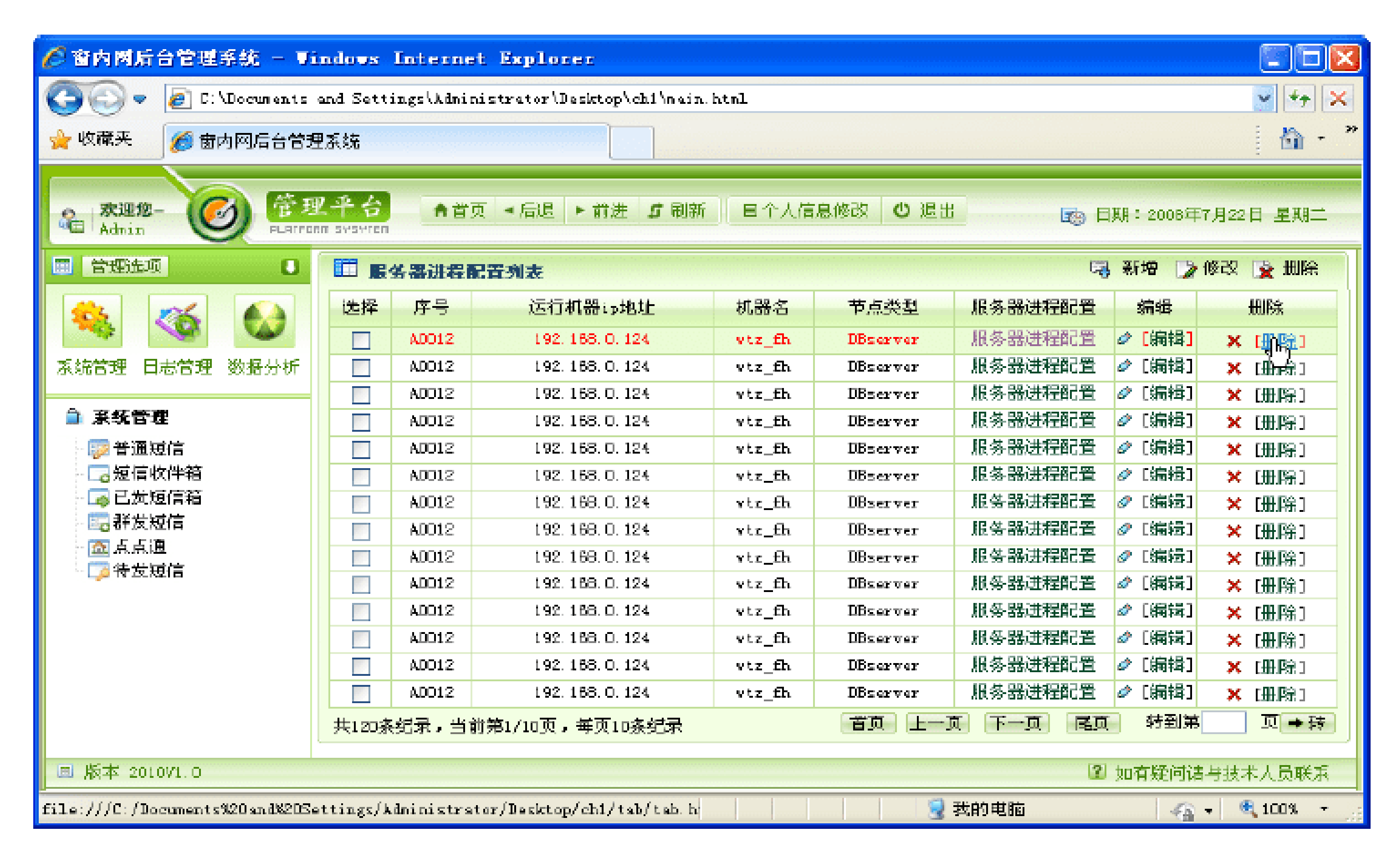


图 8-1　后台管理系统

8.1.5 实例分析



源码解析：

在上述实例中，通过设置 color 属性来设置容器内的字体颜色。同样，可以通过 background-color 属性来设置背景颜色，完成需要显示的特效。这两个属性值类型是相同的，可以使用 RGB、十六进制值进行赋值。

8.2 个 人 博 客

本节将讲解页面的背景图片设置，主要通过设置 background-image 属性、background-attachment 属性、background-repeat 属性来完成页面背景样式。下面借助个人博客案例来完成背景图片的应用。


视频教学：光盘/视频/8/8.2　个人博客.avi　　　　长度：10 分钟

8.2.1　基础知识——background 属性

在没有使用 CSS 技术之前，要为页面添加一个背景图像，通常是通过<body>标签的 background 属性来完成的。现在，已经很少使用这个属性了，它已经被 CSS 技术所替代。

background 属性综合了以上所有与背景有关的属性(即以 back-ground-开头的属性)，可以一次性地设定背景样式。格式如下：

```
background:[background-color] [background-image] [background-repeat]
    [background-attachment] [background-position]
```

其中的属性顺序可以自由调换，并且可以有选择性地设定。如果其中的某个属性没有设定属性值，那么系统会自动地为该属性添加默认值。例如，定义背景样式规则如下：

```
body
{
    background-color:Black;
    background-image:url(bk1.jpg);
    background-position:center;
    background-repeat:repeat-x;
    background-attachment:fixed
}
```

该规则等价于下面几条规则：

```
body{background:black url(bk1.jpg) center repeat-x fixed}
body{background:url(bk1.jpg) black repeat-x center fixed}
body{background:repeat-x black url(bk1.jpg) fixed center}
```

8.2.2　基础知识——background-image 属性

background-image 属性用于设定标记的背景图片。通常情况下，可以在标记<body>中应用该属性值，其效果是将图片用于整个主体网页中。



在设定背景图片时，最好同时也设定背景色，这样当背景图片因某种原因无法正常显示时，可以使用背景色来代替。当然，如果正常显示，背景图片会覆盖背景色的。

例如，在 CSS 的标记<body>中定义添加背景图片，这样每个页面都会显示该图片，详细代码如下所示：

```
<body style="background-image:url(bk2.jpg); background-color:AliceBlue">
    <p style="padding:5mm; color:Sienna">
    郑州汇智计算机技术服务有限公司成立于 2002 年。致力于把基于先进信息技术的最佳管理与业务实践普及到客户的管理与业务创新活动中，全面提供企业管理 ERP 软件、服务与解决方案，是中原地区较早的管理软件、客户关系管理软件、小型企业管理软件以及网站建设服务提供商
```

```
    </p>
</body>
```

代码中需要说明的是背景图片引用的问题，在 url 中指定引用背景图片的路径。该路径可以是图片的相对路径，也可以是绝对路径；还可以使用链接的方式链接到网络中图片的位置。运行页面，可以在浏览器中查看背景图片的显示效果。



背景图片的显示与其尺寸有关，如果图片尺寸小于浏览器可显示的范围，那么图片显示的空余部分将以空白或重复图片来填充。

背景图片适用于任何页面元素，例如，为了突出显示某一个段落，在上述代码的段落中应用背景图片，详细代码如下所示：

```
<body style="background-image:url(bk2.jpg); background-color:AliceBlue">
    <p style="padding:5mm; color:Sienna; background-image:url(1.jpg)">
郑州汇智计算机技术服务有限公司成立于 2002 年。致力于把基于先进信息技术的最佳管理与业务实践普及到客户的管理与业务创新活动中，全面提供企业管理 ERP 软件、服务与解决方案，是中原地区较早的管理软件、客户关系管理软件、小型企业管理软件以及网站建设服务提供商
    </p>
</body>
```

上述代码中引用的是相同路径下的名为 1.jpg 的图片，在浏览器中运行页面查看效果。

8.2.3 基础知识——background-attachment 属性

在一个页面中，如果使用的背景图片不能够覆盖整个页面，而只是将背景图片放置在局部位置，那么在浏览页面时，肯定会出现看不到背景图片的情况；再者，还可能出现背景图片初始可见，而随着页面的滚动又不可见。也就是说，背景图片不能时刻地随着页面的滚动而显示。

要解决上述问题，就要使用 background-attachment 属性，该属性用来设定背景图片是否随文档一起滚动。该属性包含两个属性值：scroll 和 fixed，并适用于所有元素。

- scroll：默认值，当页面滚动时，背景图片随页面一起滚动。
- fixed：背景图片固定在页面的可见区域里。

使用 background-attachment 属性，可以使背景图片始终处于视野范围内，以避免出现因页面的滚动而消失的情况。

下面两段代码是两种不同的图片显示滚动条的状态，详细代码如下所示：

```
body{
    background-image:url(my.jpg);
    background-repeat:no-repeat;
    background-attachment:scroll
}
```

上述代码中通过设置 background-attachment 属性值为 scroll 让页面滚动时，背景图片始终一起滚动。下面代码则是固定背景图片代码。

```
body{
    background-image:url(my.jpg);
    background-repeat:no-repeat;
    background-attachment:fixed
}
```

在代码中，一般来说，属性值为 fixed 其实是相对于可视范围而言的。无论页面如何滚动，图片始终处于可视范围之内，而页面中的元素内容则处于滚动状态。



最后需要说明的是，boackground-attachment:fixed 在 CSS 样式中是很常见的。但 IE 系列浏览器对其的支持是有限的。IE 7 对其提供了正确的支持，但早期版本都不能支持。尽管可以使用 JavaScript 中的 workaround 来解决这个问题，但它们带来的麻烦比其实现的价值更大一些。

8.2.4　基础知识——background-repeat 属性

background-repeat 属性用于设定背景图片是否重复平铺。该属性有 4 种属性值：repeat、repeat-x、repeat-y 和 no-repeat，默认值为 repeat。

- repeat：背景图片水平和垂直方向都重复平铺。
- repeat-x：背景图片水平方向重复平铺。
- repeat-y：背景图片垂直方向重复平铺。
- no-repeat：背景图片不重复平铺。

background-repeat 属性设置的重复背景图片是从元素的左上角开始平铺，直到水平、垂直或全部页面都被背景图片覆盖。

下面代码则是实现水平和垂直方向平铺效果的代码，如下所示：

```
body {
    background-image:url(my.jpg);
    background-repeat:repeat
}
```

在上述代码中，my.jpg 是需要设置为背景的图片。利用 background-repeat 属性设置平铺效果，接下来则是让图片平铺于页面的左边，即垂直平铺，代码如下所示：

```
body {
    background-image:url(my.jpg);
    background-repeat:repeat-y
}
```

同样，如果需要图片按水平方向平铺，则可以使用下列代码：

```
body {
    background-image:url(my.jpg);
    background-repeat:repeat-x
}
```

当然，如果背景图片符合页面需要，也可以选择不应用任何平铺方式：

```
body {
    background-image:url(my.jpg);
    background-repeat:no-repeat
}
```

在上面的 4 个样式中，为了显示各属性值的效果，没有应用背景颜色。但是，在实际应用中，在应用背景图片的同时，一定要定义背景颜色。

8.2.5　实例描述

下面将通过个人博客的实例讲解如何设置个人博客背景图片、显示博客导航和博客文章内容，以及当文章内容过多出现滚动条时背景图片和滚动条是否跟随一起滚动等功能。在本实例中主要通过使用 background-image 属性、background-attachment 属性和 background-repeat 属性来完成。

8.2.6　实例应用

【例 8-2】个人博客

首先需要设置导航和显示文章内容的布局信息，详细代码如下所示。

```
<table width="588" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td width="39"> </td>
        <td width="487"> </td>
        <td width="34"><a href="#">注册</a></td>
        <td width="28"><a href="#">登录</a></td>
    </tr>
</table>
<br/>
<table width="544" height="34" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td width="111" height="34" class="tdimag">首页</td>
        <td width="111" class="tdimag" style="color:#999">文章</td>
        <td width="111" class="tdimag">相册</td>
        <td width="111"class="tdimag">留言</td>
        <td width="100" class="tdimag">关于</td>
    </tr>
</table>
<hr/>
<br/>
<div style="font-size:14px; text-align: center;font-weight: bold">
</div>
```

上述代码中，定义了【注册】和【登录】两个超链接。接下来要对页面添加背景图片以及设置 td 的背景图片使它具有导航功能，详细代码如下所示：

```
<style type="text/css">
body{
    background-image:url(images/http imgload13.jpg);
    background-repeat:no-repeat;
    background-attachment:fixed;
    font-size:12px;
    }
    .tdimag{
        background-image:url(images/bqqh2.jpg);
        background-repeat:no-repeat;
        text-align: center;
        }
        a:link {
    color: #CFF;
}
</style>
```

上述代码通过设置 body 元素的 background-image 属性来设置页面的背景图片，使用 background-repeat 属性来控制图片的重复方式，设置 background-attachment 属性值为 fixed 使出现滚动条时背景图片不随滚动条一起滚动。

8.2.7　运行结果

保存修改页面运行 index.html 页面，运行结果如图 8-2 所示。

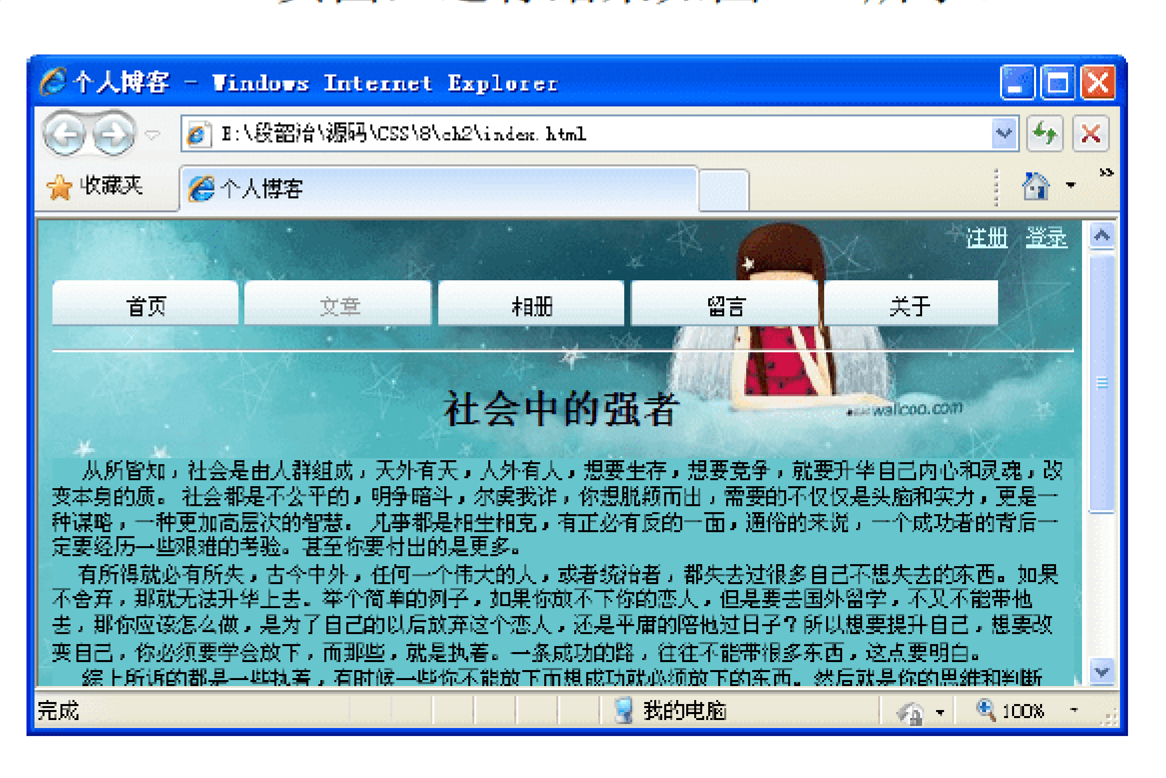


图 8-2　个人博客

8.2.8　实例分析

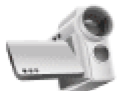


源码解析：

设置页面背景图片只需要设置 background-image 属性，通过设置 background-repeat 属性，可以控制页面背景图像的平铺方式，repeat 则是图片水平和垂直方向都重复平铺；而设置 background-attachment 属性则是控制背景图片和滚动条是否一起滚动。

8.3　设置网站 Logo 位置

随着技术的不断更新，人们的需求也在不断地增加。有的时候在开发项目中可能会遇到这样的问题：要求在页面内显示一张背景图片，而却要求背景图片的位置固定到某一个地方。为了方便开发者的使用，CSS 提供了很多关于设置背景图片位置的属性，接下来将对 background-position 属性、background-position-y 属性、background-position-x 属性、layer-background-color 属性和 layer-background-image 属性进行详细讲解。

**视频教学：** 光盘/视频/8/8.3　设置网站 logo 位置(1)和 8.3　设置网站 logo 位置(2).avi
长度：8 分钟

8.3.1　基础知识——background-position 属性

background-position 属性用于指定背景图片在页面中所处的位置。该属性的属性值可以分为 4 类：绝对定义位置(length)、百分比定义位置(percentage)、垂直对齐值和水平对齐值。其中垂直对齐值包括 top、center 和 bottom，水平对齐值包括 left、center 和 right。

- length：设置图片与边距水平、垂直方向的距离，长度单位为 cm、mm、px 等。
- percentage：以页面元素框的宽度或高度的百分比放置图片。
- top：背景图片顶部显示。
- center：背景图片居中显示。
- bottom：背景图片底部显示。
- left：背景图片左部居中显示。
- right：背景图片右部居中显示。



由于 background-position 属性是描述背景图片的位置的，所以必须与 background-image 属性同时使用。

另外，垂直对齐值还可以与水平对齐值一起使用，从而决定图片的垂直位置和水平位置，例如下列样式代码：

```
body {
    background-image:url(jt.gif);
    background-repeat:no-repeat;
    background-position:top right
}
```

上述代码设置图片位置为上端靠右，通过设置 background-position 属性值为 top right 即可实现。

使用垂直对齐值和水平对齐值只能格式化放置图片，如果在页面中要自由定义图片的位置，则需要使用确定数值或百分比，详细代码如下所示：

```
body {
    background-image:url(3.jpg);
    background-repeat:no-repeat;
    background-position:20% 20%
}
body {
    background-image:url(3.jpg);
    background-repeat:no-repeat;
    background-position:20px 30px
}
```

第一条规则表示图片的垂直位置与水平位置均相对于页面的 20%放置，所以页面的大小决定了图片所处的位置；第二条规则表示图片的水平距离为 20px，垂直距离为 30px，该规则定义了图片的绝对位置。在浏览器的<body>标记中可以分别使用两条规则。



background-position 属性适用于 XHTML 中的块标记及可替换标记，比如 img、input、select 等。

8.3.2　基础知识——background-position-x 属性

background-position-x 属性用于设置或检索对象的背景图像横坐标位置，必须先指定 background-image 属性，默认值为 0%。该属性不受对象的补丁属性(padding)的影响，它有 4 个参数值。

- length：由浮点数字和单位标识符组成的长度值。
- left：居左。
- center：居中。
- right：居右。

下面通过一段代码来介绍如何使用 background-position-x 属性来完成图片坐标的更改，详细代码如下所示：

```
<style type="text/css">
p{
    background-Position-x:35%
    background-repeat:no-repeat;
    height:80px
}
</style>
```

上述代码设置图片在当前容器中的 x 坐标的百分比，从而控制图片在页面中显示的位置。

8.3.3　基础知识——background-position-y 属性

background-position-y 属性用于设置或检索对象的背景图像纵坐标位置。该属性和 background-position-x 属性使用方法类似，唯一不同的就是它的显示方式不同，

background-position-y 属性也有 4 个参数。

- length：由浮点数字和单位标识符组成的长度值。
- top：居顶。
- center：居中。
- bottom：居底。

background-position-y 属性是用来控制图片居于顶部、中部或底部的位置，同样还可以通过设置百分比来控制图片的位置，详细代码如下所示：

```
<style type="text/css">
p{
    background-image: url("images/~~Logo.gif");
    background-position-y:bottom;
    background-repeat:no-repeat;
    height:80px
}
</style>
```

上述代码中，同样设置 background-image 属性指定一张图片，在这里设置 background-position-y 的值为 bottom，则是为了使图片能够显示在最底部。

8.3.4 基础知识——layer-background-color 属性

layer-background-color 属性用来设置或检索对象整个区域的背景颜色，该属性有两个参数值。

- transparent：背景色为透明。
- color：为背景指定颜色。

下面这段代码则是如何使用该属性的基本语法规则：

```
<style type="text/css">
div {
layer-background-color: transparent;
}
</style >
```

上述代码是对 layer-background-color 属性进行赋值，让 div 元素的背景色为透明。

8.3.5 基础知识——layer-background-image 属性

layer-background-image 属性用来设置或检索对象整个区域的背景图像，该属性有两个参数值。

- none：无背景图。
- url：使用绝对或相对地址指定背景图像。

下面实例是对 layer-background-image 属性使用方法的具体讲解，详细代码如下所示：

```
<style type="text/css">
code {
position: absolute;
top: 100px;
left: 300px;
width: 200px;
border: thin solid black;
background-image: url("comet.jpg");
layer-background-image: url("bb_comet.jpg");
}
</style >
```

8.3.6　实例描述

在互联网领域中，无论网站的规模是大还是小，是知名度高的还是默默无闻的，它们通常都具有一个共同的特点，就是每个网站都有自己独特的网站 Logo，它代表网站的形象，几乎所有的网站 Logo 都位于页面的左上角。下面通过设置网站 Logo 位置的案例来进行讲解。

8.3.7　实例应用

【例 8-3】设置网站 Logo 位置

首先需要设置页面内容，将 Logo 作为页面的背景图片显示在页面内，通过设置相关的属性使背景图片显示在相应的位置，详细代码如下所示：

```
body {
    background-image:url(Logo.jpg);
    background-repeat:no-repeat;
    layer-background-color: transparent;
    background-position:3% 8%
}
```

在上述代码中，首先指定了 background-image 为网站 Logo。然后以背景图片的形式显示在页面中。接着则是设置背景图片不发生改变，图片不重复平铺。这样就可以将一个完整的图片显示出来。接下来需要将 layer-background-color 的属性值设置为 transparent，使其背景色显示为透明的效果。background-position 属性是设置背景图片位置比较重要的属性，该属性既可使用数字表示也可使用百分数，可以根据具体情况灵活运用。

8.3.8　运行结果

保存修改后的页面代码，在浏览器中运行 indexa.html 页面的结果如图 8-3 所示。



图 8-3　网站 Logo

8.3.9　实例分析



源码解析：

通过上述实例的讲解，在使用 background-position 属性时，可以根据其属性值的不同来改变背景图片在页面中显示的位置，该属性有顶部居中显示、居中显示、底部居中显示、左部居中显示和右部居中显示等选项，如果这些显示位置还不能够满足需求，还可以选择使用 percentage 的属性值以页面元素框的宽度或高度的百分比来放置图片。

8.4　常见问题解答

8.4.1　背景图片重复显示



设置背景图片为什么会重复显示？
网络课堂：http://bbs.itzcn.com/thread-11775-1-1.html

在使用 background-image 属性设置页面背景图片时，发现图片是重复显示出来的，代码如下所示：

```
p{
    background-image: url("images/~~Logo.gif");
    background-positionY:center;
    height:80px
}
```

在上述代码中，设置 background-image 属性的 url 值为图片路径，当在浏览器中运行这些代码时就出现问题了。

【解决办法】背景图片的显示与其尺寸有关，如果图片尺寸小于浏览器可显示的范围，那么图片显示的空余的部分将以空白或重复图片来填充，可以设置 background-repeat 属性来控制

显示图片的样式。

8.4.2　背景滚动问题



使用 background-attachment 属性没有效果！
网络课堂：http://bbs.itzcn.com/thread-11776-1-1.html

在制作个人博客的时候，需要内容和背景图片不一起滚动，因此我就将 background-attachment 属性的值设置为 fixed，然后使用 IE 浏览器打开时内容和背景图片却还是一起滚动，请问这是什么原因？

【解决办法】boackground-attachment:fixed 在 CSS 样式是很常见的，但 IE 系列浏览器对它的支持是有限的。IE7 对其提供了正确的支持，但早期的版本都不能支持。尽管可以使用 JavaScript 中的 workaround 技术来解决这个问题，但它们带来的麻烦比其实现的价值更大一些。

8.5　习　　题

1. 填空题

(1) CSS 中为开发人员预设了__________种颜色。

(2) 在 CSS 中颜色值可以分为十六进制值、颜色简称和__________三种。

(3) __________属性用来设置或检索对象的背景颜色。

(4) 通常情况下，可以使用 CSS 中的__________属性来设置页面的背景图片。

(5) background-repeat 属性用于设定背景图片是否重复平铺，该属性中的__________值用来设置背景图片不重复平铺。

(6) 由于 background-position 属性是描述背景图片的位置的，所以必须与__________属性同时使用。

(7) background-position-x 属性用来设置或检索对象的背景图像的横坐标位置，该属性中 center 值的作用是__________。

2. 选择题

(1) 下列选项中属于 background-attachment 属性的值的是__________。

A. none　　B. color　　C. fixed　　D. center

(2) 下列选项都是 background-position 属性值的一些特性，其中是 bottom 属性值的作用是__________。

A. 背景图片居中显示　　B. 背景图片右部居中显示
C. 背景图片顶部居中显示　　D. 背景图片底部居中显示

(3) layer-background-color 属性中的__________属性值用来设置背景色透明。

A. tsransparent　　B. url　　C. none　　D. color

(4) background-position 属性用于指定背景图片在页面中所处的位置，该属性的属性值可

以分为 4 类，下面不属于该属性的值的是__________。

A. 十六进制值　　　　　　B. 百分比定义位置

C. 垂直对齐值　　　　　　D. 水平对齐值

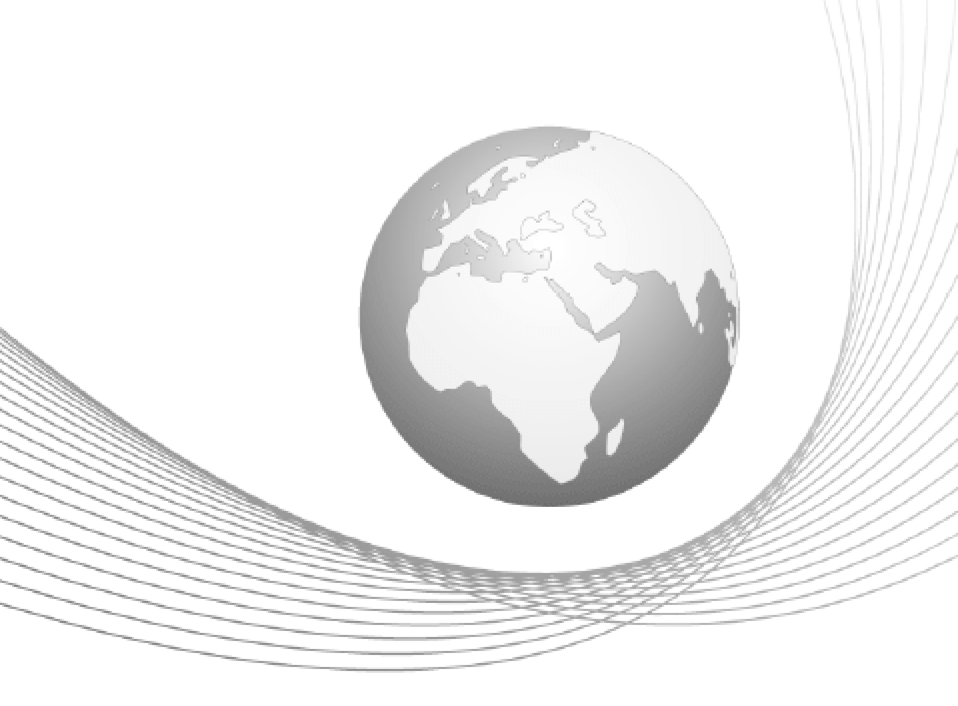
3. 上机练习

上机练习：窗内网首页设计

实例要求：在页面的左上角添加网站 Logo，同时设置登录窗口的字体颜色和背景颜色，当出现滚动条时页面内容和背景图片不同时滚动。页面背景图片水平和垂直方向都重复平铺。运行结果如图 8-4 所示。



图 8-4　窗内网首页面



第9章 列表属性

内容摘要：

列表是 XHTML 中一种很有用的显示方式，它可以把相关的具有并列关系的内容整齐地垂直排列，不仅很好地归纳了内容，而且使页面也显得整洁，增强了页面的条理性。而在 CSS 中为控制列表提供了符号列表、图像列表和位置列表三种样式。本章将对这三种样式以及包含这三种样式的集合样式属性进行详细介绍。

学习目标：

- 掌握使用 list-style 属性的方法
- 掌握使用 list-style-image 属性的方法
- 掌握使用 list-style-position 属性的方法
- 掌握使用 list-style-type 属性的方法
- 掌握使用 marker-offset 属性的方法

9.1 构建试卷页面

在 Web 开发中，很多情况下需要给列表项定义项目符号。在 CSS 中为列表项定义项目符号，可以使用 list-style-type 属性。本节将为大家讲述 CSS 中的符号列表属性——list-style-type。

**视频教学：**光盘/视频/9/9.1 构建试卷页面.avi　　长度：5 分钟

9.1.1 基础知识——list-style-type 属性

在 CSS 中可以使用 list-style-type 属性为列表项定义项目符号。该属性值说明如表 9-1 所示。

表 9-1 list-style-type 属性值

属性值	说明	属性值	说明
Disc	实心圆形项目符号	circle	空心圆形项目符号
decimal	十进制数项目符号，如：1、2、3…	decimal-leading-zero	十进制数项目符号，不足两位的补齐前导 0，例如：01、02、03…
lower-roman	小写罗马文字项目符号，如：i、ii、iii…	upper-roman	大写罗马文字项目符号，如：I、II、III…
lower-greek	小写希腊字母项目符号，如α(alpha)、β(beta)、γ (gamma)…	square	实心方形项目符号
lower-latin	小写拉丁文项目符号，如 a、b、c…	upper-latin	大写拉丁文项目符号，如 A、B、C…
armenia	亚美尼亚数字项目符号	georgian	乔治亚数字项目符号例如：an、ban、gan…
lower-alpha	小写拉丁文项目符号，如 a、b、c…	upper-alpha	大写拉丁文项目符号，如 A、B、C…
none	无项目符号	inherit	继承

list-style-type 属性只能够用于 XHTML 中的列表元素，比如，ul、ol 等。而且列表项元素 li 能够单独定义列表符号，例如：

```
<body>
    <h1>图书销量排行榜</h1>
    <ol>
        <li style="list-style-type:disc">《世界博物馆巡礼系列》</li>
        <li style="list-style-type:circle">《老照片》</li>
        <li style="list-style-type:square">《中华传统文化经典文库》</li>
        <li style="list-style-type:decimal">《鼠疫》</li>
        <li style="list-style-type:upper-alpha">《我把青春献给您》</li>
        <li style="list-style-type:lower-alpha">《经济学家茶座》</li>
```

```
        <li style="list-style-type:upper-roman">《红楼小讲——大家小书》</li>
        <li style="list-style-type:lower-roman">《哈耶克传》</li>
        <li style="list-style-type:none">《大地玫瑰包系列》</li>
    </ol>
</body>
```

在 IE 浏览器中运行页面，效果如图 9-1 所示。

从上面的一个案例中，可以看出 CSS 不区分无序与有序列表项，所以即使是有序列表项也可以使用前三种属性值定义项目符号。如下面代码定义：

```
<body>
    <h1>图书销量排行榜</h1>
    <ol style="list-style-type:square;">
        <li>《世界博物馆巡礼系列》</li>
        <li>《老照片》</li>
        <li>《中华传统文化经典文库》</li>
    </ol>
</body>
```

在 IE 浏览器中查看一下，效果如图 9-2 所示。

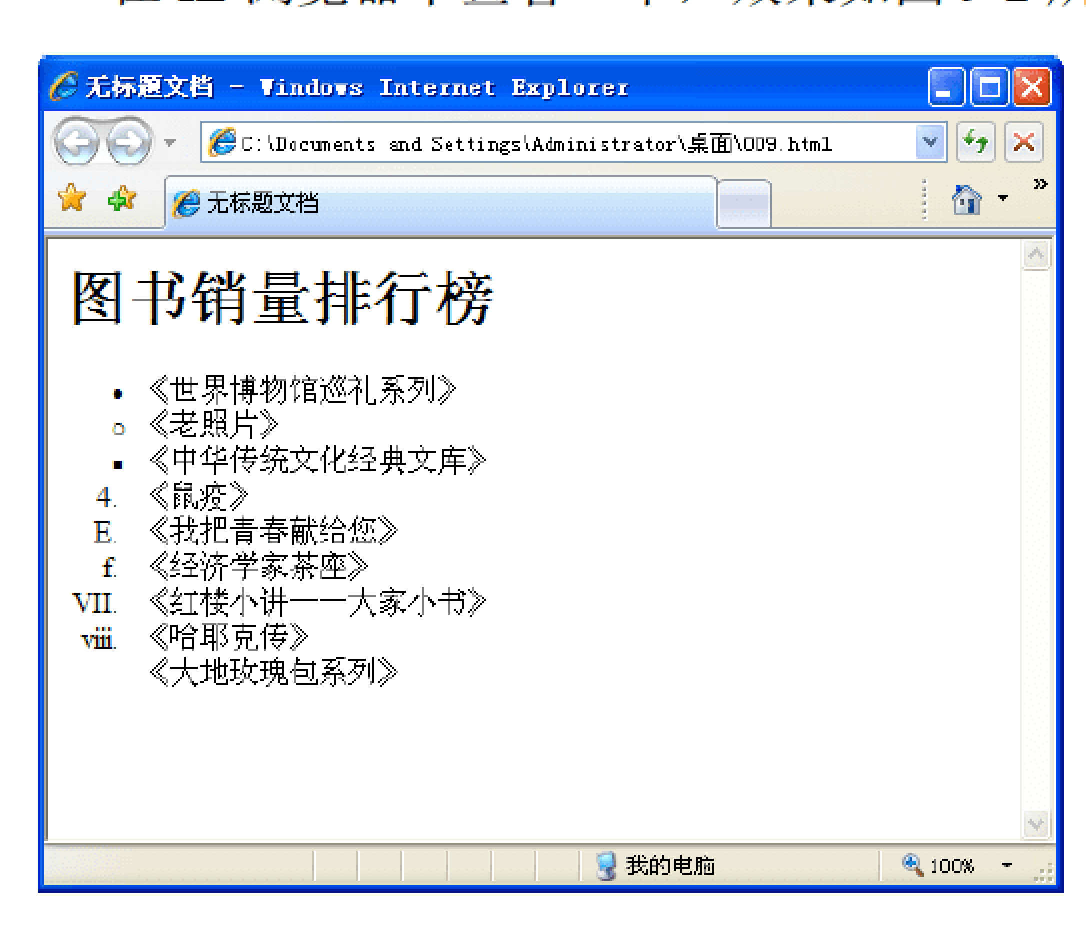


图 9-1　列表项

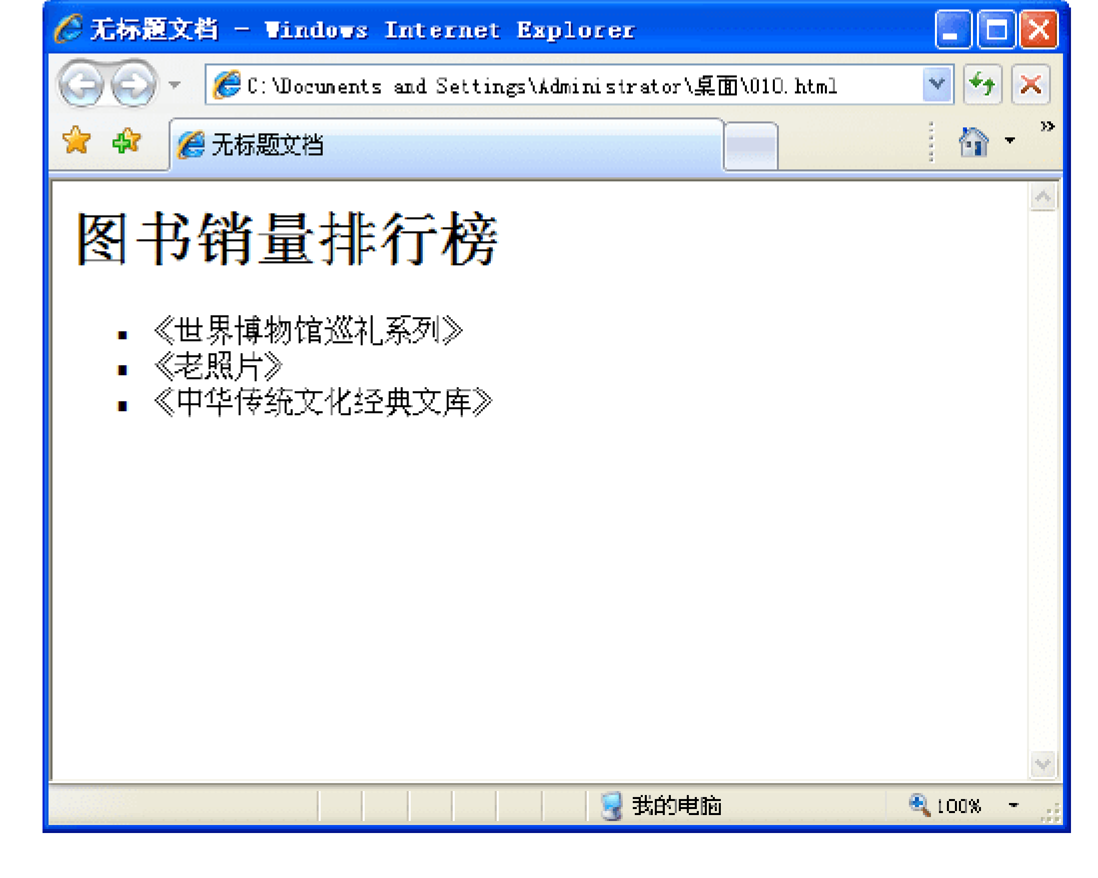


图 9-2　有序列表项



如果同时在 ol 标签中和 li 标签中定义了不同的项目符号样式，那么 li 所表示的列表项将按其定义样式显示项目符号，其余的 li 列表项将按 ol 标签中定义的样式显示项目符号。

list-style-type 属性的默认值是 disc，因此在没有特殊声明的情况下，所有列表(无序或有序)都将使用实心圆形作为列表项的项目符号。

如果在某个列表项前(li 元素)不需要项目符号，则可以设定属性值为 none。但是这并不影响上级元素(ol 标记)设定的有序项目符号的排列。代码如下所示：

```
<body>
    <h1>图书销量排行榜</h1>
    <ol style="list-style-type:decimal;">
```

```
        <li>《世界博物馆巡礼系列》</li>
        <li>《老照片》</li>
        <li>《中华传统文化经典文库》</li>
        <li style="list-style-type:none">《大地玫瑰包系列》</li>
        <li>《鼠疫》</li>
        <li>《我把青春献给您》</li>
        <li style="list-style-type:none">《经济学家茶座》</li>
        <li>《红楼小讲——大家小书》</li>
        <li>《哈耶克传》</li>
    </ol>
</body>
```

在 IE 浏览器中查看页面，效果如图 9-3 所示。



图 9-3　自动排列计数



list-style-type 属性是可以继承的，如果想要定义新的符号列表，则需要单独定义。

9.1.2　实例描述

定义符号列表的样式种类很多，而每种样式都有它的用处。比如说页面中要罗列出一些题目，需要用到十进制数项目符号；页面中需要显示每道题的选择答案，需要用到大写字母项目符号或者小写字母项目符号……而同一张页面中也许需要出现很多的列表，那么怎样合理地使用符号列表属性呢？下面，我们就来尝试一下吧！

9.1.3　实例应用

【例 9-1】构建试卷页面

(1) 新建一个 HTML 页面，命名为 9-1.html。

(2) 在页面中定义嵌套列表，代码如下所示：

```
<ul>
    <li>CSS 是利用什么 XHTML 标记构建网页布局？()</li>
      <ul id="upperalpha">
        <li>dir</li>
          <li>div</li>
          <li>dis</li>
          <li>dif</li>
      </ul>
      <li>在 CSS 语言中下列哪一项是“左边框”的语法？()</li>
      <ul id="upperalpha">
        <li>border-left-width</li>
          <li>border-top-width</li>
          <li>border-left</li>
          <li>border-top-width</li>
      </ul>
      <li>在 CSS 语言中下列哪一项的适用对象是“所有对象”？()</li>
      <ul id="upperalpha">
        <li>背景附件</li>
          <li>文本排列</li>
          <li>纵向排列</li>
          <li>文本缩进</li>
      </ul>
      <li>下列选项中不属于 CSS 文本属性的是()。</li>
      <ul id="upperalpha">
        <li>font-size</li>
          <li>text-transform</li>
          <li>text-align</li>
          <li>line-height</li>
      </ul>
      <li>在 CSS 中不属于添加在当前页面的形式是()。</li>
      <ul id="upperalpha">
        <li>内嵌式样式表</li>
          <li>嵌入式样式表</li>
          <li>层迭式样式表</li>
          <li>链接式样式表</li>
      </ul>
   </ul>
```

(3) 定义上面的嵌套列表样式表，代码如下：

```
ul
{
    font-size:18px;
    font-weight:bold;
    list-style-type:decimal;
```

```
    }
    ul #upperalpha
    {
        font-size:16px;
        font-weight:normal;
        list-style-type:upper-alpha;
    }
```

(4) 保存 9-1.html 页面。

9.1.4　运行结果

在 IE 浏览器中运行 9-1.html 页面，出现如图 9-4 所示的页面效果。



图 9-4　使用 list-style-type 属性构建试卷页面

9.1.5　实例分析



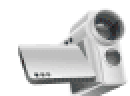
源码解析：

在上面的实例中，定义列表的 list-style-type 属性默认值为 decimal(十进制数项目符号)，并在其每个列表项下面嵌套无序列表(ul)。定义嵌套列表的 list-style-type 值为 upper-alpha(大写字母项目符号)，从而在页面中显示出仿试卷模式的页面。

9.2　新闻中心图像列表

为了使页面更加生动，除了使用预置的项目符号外，还可以使用图像作为项目符号，用来标识各列表项。CSS 中使用 list-style-image 属性来实现使用图像替代列表项的常规项目符号。

这节将为大家讲述如何使用 list-style-image 属性来定义列表项的项目符号。



视频教学：光盘/视频/9/9.2　新闻中心图像列表.avi　　　　长度：4 分钟

9.2.1　基础知识——list-style-image 属性

在 CSS 中，可以使用 list-style-image 属性来实现图像列表。使用 list-style-image 属性为列表项指定图像很简单，只要在属性中指定图像的路径就可以了，格式如下：

```
list-style-image:url(图像路径)/none
```

list-style-image 属性可以取两个值。

- none：没有替换的图片。
- url(图像路径)：要替换图片的路径。

例如，下面的代码指定了一个图像作为各列表项的项目符号。

```
<body>
    <h1>图书销量排行榜</h1>
   <ol style="list-style-image:url(images/emot-20.gif);">
    <li>《世界博物馆巡礼系列》</li>
       <li>《老照片》</li>
       <li>《中华传统文化经典文库》</li>
   </ol>
</body>
```

在 IE 浏览器中运行页面，效果如图 9-5 所示。



引用图像的路径可以使用相对路径，也可使用绝对路径，或者指定网络资源链接。

使用图像项目符号，有时会出现图像不能够正常显示的情况，在这种情况下，最好的办法就是在定义图像项目符号的同时，设置符号列表。这样能够保证在图像不能够正确加载时，使用其他项目符号来代替，如下面的代码：

```
<body style="background-image:url(images/http_imgload7.jpg);">
    <h1>图书销量排行榜</h1>
   <ol style="list-style-image:url(images/emot-21.gif);
list-style-type:circle;">
    <li>《世界博物馆巡礼系列》</li>
       <li>《老照片》</li>
       <li>《中华传统文化经典文库》</li>
   </ol>
</body>
```

如果上面代码中的 images 文件夹下不存在“emot-21.gif”图片时，在浏览器中浏览效果，如图 9-6 所示，显示的是空心圆形的项目符号。



图 9-5　图像项目符号



图 9-6　使用预置项目符号

9.2.2　实例描述

你做过新闻管理系统吗？在前台展示中，是否需要把新闻展现在页面中，并且需要在每条新闻的前面加上一张标识“NEW”的图片呢？学习了 CSS 中的图像列表属性，这些对大家来说就是轻而易举的事情了，并非需要在页面中编辑大量的代码来完成这项工作，只需要一行 CSS 代码就可以了。

9.2.3　实例应用

【例 9-2】 新闻中心图像列表

(1) 新建 HTML 页面，命名为 9-2.html。

(2) 在页面中定义第一个“ol”列表，设置 list-style-image 属性值为一张存在的图片，并设置 list-style-type 的属性值，代码如下：

```
    <h1>新闻中心</h1>
        <ol style="list-style-image:url(images/sucaiwcom70930000ir.gif); list-style-type:square;">
            <li>专家建议国考公开计分标准和答案，以方便考生……</li>
            <li>好心带讨饭者回家，没想到反被他们抢了</li>
            <li>新 107 国道 15 日启用</li>
            <li>郑州明年有望用垃圾发电</li>
            <li>办离婚手续起争议</li>
            <li>煤化建业公司 11 亿元拿地 86 亩</li>
            <li>杭州回应名人入住“别墅保障房”</li>
            <li>官员遭押 590 天后被判无罪，官复原职</li>
        </ol>
```

(3) 接着在 9-2.html 页面中定义第二个“ol”列表，设置 list-style-image 属性值为一张实际不存在的图片，并设置 list-style-type 的属性值，代码如下：

```
<h1>房产动态</h1>
    <ol style="list-style-image:url(images/none.jpg); list-style-type:square;">
        <li>商业地产抗衡住宅地产</li>
        <li>“最牛钉子户”占道七年</li>
        <li>70 亿富婆身陷“别墅保障房”风波</li>
        <li>中科院“住房绿皮书”： 邕房价泡沫成分 50%</li>
    </ol>
```

9.2.4　运行结果

在 IE 浏览器中运行 9-2.html 页面，展示新闻动态。效果如图 9-7 所示。

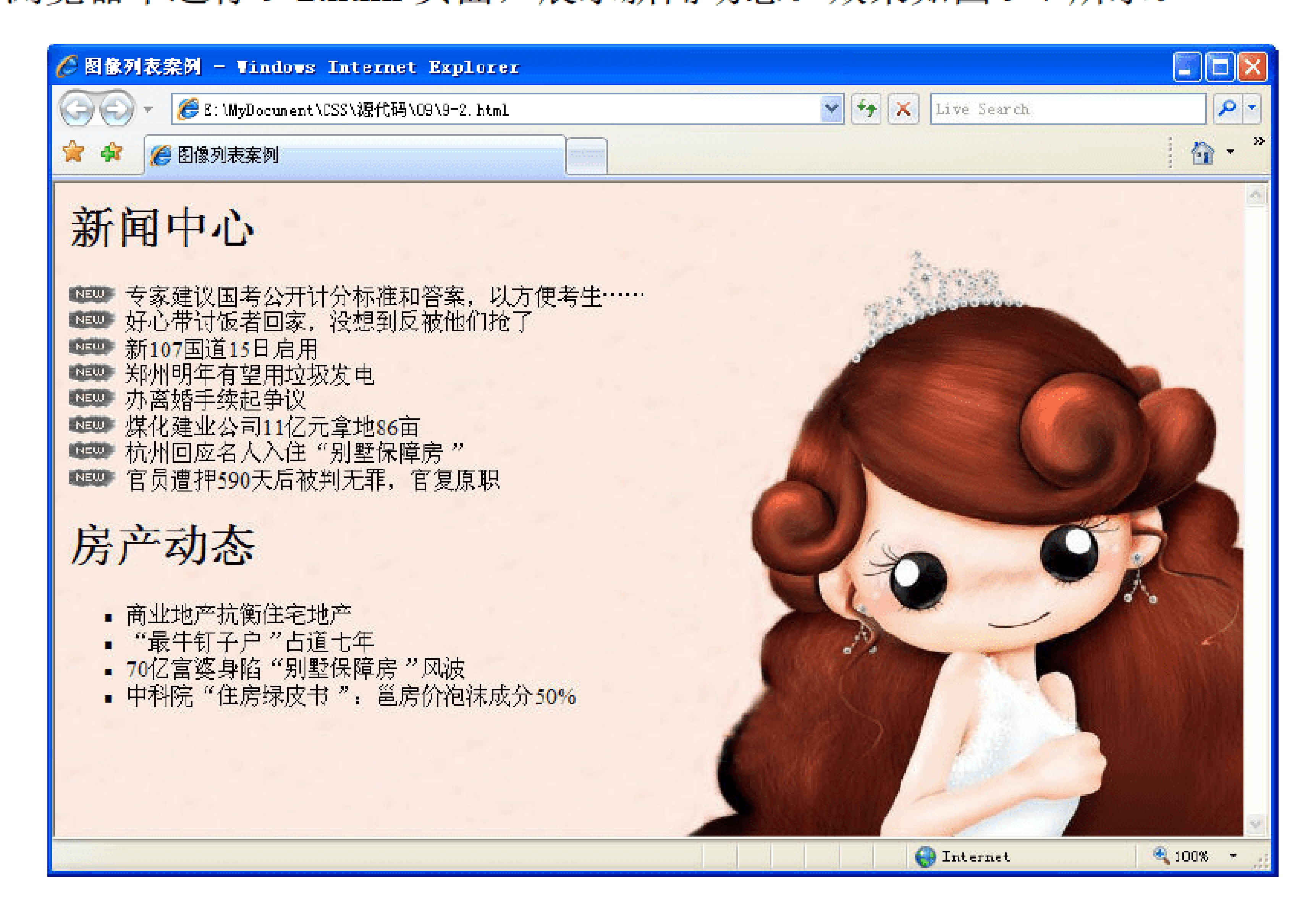


图 9-7　新闻中心图像列表

9.2.5　实例分析

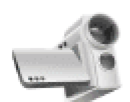


源码解析：

在上面的实例中，定义了“新闻中心”列表(ol)和“房产动态”列表(ol)，并定义了各个列表的 list-style-image 属性，其中，“新闻动态”列表的 URL 地址是有效的，因此，在“新闻中心”列表项中显示出了图像，而非实心方形项目符号。对于“房产动态”列表中的 list-style-image 属性值的 URL 是无效的，因此，采用 list-style-type 属性值作为项目符号来显示。

9.3 “阶梯形”新闻展示页面

在页面布局时，有时需要将项目符号放入到列表项内部，而有时需要将项目符号突出于列表项内容，这时就可以使用 CSS 的 list-style-position 属性，该属性可以用来改变项目符号与列表内容的位置。本节将为大家讲述如何使用 list-style-position 属性设置列表项的位置。

视频教学：光盘/视频/9/9.3 “阶梯形”新闻展示页面.avi 长度：11 分钟

9.3.1 基础知识——list-style-position 属性

list-style-position 属性可以设置在何处放置列表项符号，该属性用于声明列表符号相对于列表项内容的位置，它的有效值如表 9-2 所示。

表 9-2 list-style-position 属性值

属 性 值	说 明
inside	将项目符号放入列表项内部
outside	将项目符号突出于列表项内容
inherit	规定应该从父元素继承 list-style-position 属性的值

list-style-position 属性的有效值有三个，分别是 inside、outside 和 inherit，默认值为 outside。下面给出一个实例，来说明它的属性值所表示的列表项位置的不同。

```
 <h1>水果简介</h1>
<ol style="list-style-type:circle">
 <li style="list-style-position:inside">
        吃香蕉治忧郁症。过度用脑导致人体内维生素、矿物质及热量缺乏，除了大脑疲惫，还常常感
到情绪低落。此时补充香蕉可提供所需营养物质并缓解消极情绪。由于过度用脑消耗多种维生素，因此
营养师建议同时补充善存等多维生素片。
      </li>
      <li style="list-style-position:outside">
        猕猴桃治牙龈出血。牙龈健康与维生素 C 息息相关。缺乏维生素 C 的人牙龈变得脆弱，常常
出血、肿胀，甚至引起牙齿松动。猕猴桃的维生素 C 含量是水果中最丰富的，因此是最有益于牙龈健康
的水果。
      </li>
</ol>
```

在代码中，第一个列表项样式定义将项目符号放入列表项内部，而第二个列表项样式定义将项目符号突出于列表项内容。在 IE 浏览器中运行页面，查看效果如图 9-8 所示。

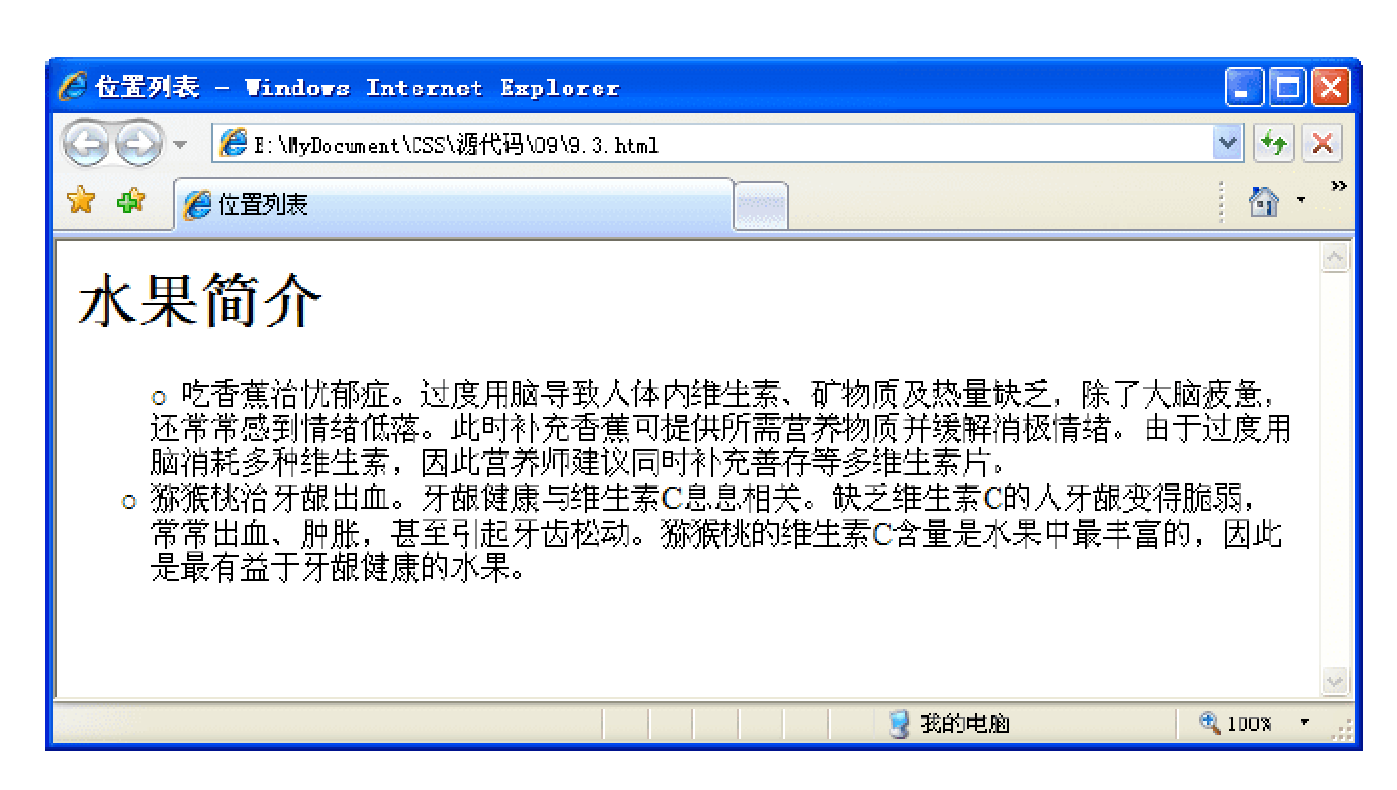


图 9-8　列表项位置

9.3.2　实例描述

新闻展示页面的排版“千姿百态”，甚至可以是“阶梯形”的，每个栏目下的新闻都要比上面一个栏目的新闻项多出几个字符的位置。像这样的排版，完全可以使用 list-style-position 属性来完成，下面就来做一个“阶梯形”的新闻展示页面吧！

9.3.3　实例应用

【例 9-3】制作“阶梯形”新闻展示页面

(1) 新建 HTML 页面，命名为 9-3.html。

(2) 在页面中定义新闻中心的“ul”列表，并命名 id 的属性值为“inside”，代码如下：

```
 <h1>新闻中心</h1>
<ul id="inside">
 <li>专家建议国考公开计分标准和答案，以方便考生……</li>
    <li>好心带讨饭者回家，没想到反被他们抢了</li>
    <li>新 107 国道 15 日启用</li>
    <li>郑州明年有望用垃圾发电</li>
    <li>办离婚手续起争议</li>
    <li>煤化建业公司 11 亿元拿地 86 亩</li>
    <li>杭州回应名人入住“别墅保障房”</li>
    <li>官员遭押 590 天后被判无罪，官复原职</li>
</ul>
```

(3) 接着在页面中定义娱乐新闻的“ul”列表，它是一个普通的列表，代码如下：

```
<h1>娱乐新闻</h1>
<ul>
 <li>谁在整容后华丽脱身</li>
    <li>超女们加油</li>
</ul>
```

(4) 继续在页面中定义房产动态的“ul”列表，并定义 id 的属性值为“outside”，代码

如下：

```
<h1>房产动态</h1>
<ul id="outside">
 <li>商业地产抗衡住宅地产</li>
    <li>“最牛钉子户”占道七年</li>
    <li>70 亿富婆身陷“别墅保障房”风波</li>
    <li>中科院“住房绿皮书”：矣房价泡沫成分 50%</li>
</ul>
```

(5) 在页面中定义 id 为“inside”的“ul”列表的列表项样式表，代码如下：

```
ul#inside
{
    list-style-position: inside;
    list-style-image:url(images/70040000iq.gif);
}
```

(6) 最后在页面中定义 id 为“outside”的“ul”列表的列表项样式表，代码如下：

```
ul#outside
{
    list-style-position: outside;
    list-style-image:url(images/10930000fq.gif);
}
```

9.3.4 运行结果

【例 9-4】“阶梯形”的新闻展示页面

在 IE 浏览器中运行 9-3.html 页面，出现如图 9-9 所示的界面效果。



图 9-9　“阶梯形”的新闻展示页面

9.3.5　实例分析



源码解析：

在上面的实例中定义了三个列表，第一个列表项样式定义将项目符号放入列表项内部，第二个列表项样式定义为正常模式，而第三个列表项样式定义将项目符号突出于列表项内容，从而形成了“阶梯形”。其实，大家可以将第一个列表项和第三个列表项分别与第二个列表项做比较，第二个列表项是正常模式，这样就能很清楚地了解到 list-style-position 属性值 inside 和 outside 的区别了。

9.4　制作文章详解页面

上面介绍的三种列表样式属性，其实可以将它们集合到一个属性中，这个属性就是 list-style 属性，那么如何使用这个属性呢？使用这个属性设置列表能起到什么样的效果呢？本节将为大家讲述 list-style 的相关内容。

**视频教学：**光盘/视频/9/9.4　制作文章详解页面.avi　　长度：5 分钟

9.4.1　基础知识——list-style 属性

list-style 属性是一种同时设置标签及其位置的简略方式，它的值包括 list-style-type、list-style-image 和 list-style-position 属性的所有合法值，其格式如下：

```
list-tyle:[list-style-type] [list-style-position][list-style-image]
```

1．按顺序设置属性

可以按顺序设置如下三个属性：

- list-style-type；
- list-style-position；
- list-style-image。

例如：

```
li{circle inside url(p9.jpg);}
```

2．存在空缺属性

对于 list-style-tyle 属性、list-style-position 属性和 list-style-image 属性，可以出现空缺，空缺的属性将使用默认值替代，例如：

```
li{circle inside;}
```

3．强制设置属性

可以强制设定 list-style-type 属性、list-style-position 属性和 list-style-image 属性的值，而不采用 list-style 属性值的默认排序，例如：

```
li{list-style-type:circle; list-style-position:inside;
list-style-image:url(p9.jpg);}
```

这样设置，三个值的顺序可以任意调换，并且也可以出现空缺，空缺的属性将使用默认值替代。例如，下面的两条规则是等价的。

```
li{
    list-style-image:url(A001161.jpg);
}
li{
    list-style-image:url(A001161.jpg);
    list-style-type:disc;
    list-style-position:outside;
}
```

有一点需要注意，如果既定义了 list-style-type 属性，又定义了 list-style-image 属性，那么无论哪个属性在前，将会以 list-style-image 属性定义的列表样式出现，如下列两条样式规则所显示的效果是相同的：

```
ol{
    list-style-type:circle;
    list-style-image:url(p9.jpg);
    list-style-position:outside;
}
ol{
    list-style-image:url(p9.jpg);
    list-style-type:circle;
    list-style-position:outside;
}
```

在 IE 浏览器中运行 9.4.html，效果如图 9-10 所示。



图 9-10　列表样式显示

9.4.2　实例描述

使用 list-style 属性，可以创建每一级具有不同编号形式的嵌套列表。假定希望创建一组三个嵌套的列表，第一个列表的标签是 decimal：1、2、3…；第一个嵌套列表的标签是 upper-alpha：A、B、C…；第二个嵌套列表的标签是 upper-roman：I、II、III…。那么，下面我们就来制作一个具有不同编号形式的嵌套列表吧！

9.4.3　实例应用

【例 9-5】制作文章详解页面

(1) 新建 HTML 文件，命名为 9-4.html。

(2) 编辑 9-4.html 页面，在页面中定义三个嵌套列表，其详细代码如下：

```
<h1 align="center">文章讲解</h1>
<div id="divol">
    <p>
        list-style 属性是一种同时设置标签及其位置的简略方式，它的值包括
list-style-type、        list-style-image 和<br />list-style-position 属性的所有
合法值。
    </p>
    <ol>
        <li>按顺序设置属性</li>
    <li>存在空缺属性</li>
    <ol>
        <li>list-style-type 空缺</li>
        <li>list-style-position 空缺</li>
        <ol>
            <li>默认值为 outside</li>
            <li>设置属性值为 inside</li>
        </ol>
        <li>list-style-image 空缺</li>
        </ol>
        <li>强制设置属性</li>
    </ol>
</div>
```

(3) 定义三个嵌套列表的列表项样式，即第一个列表项的项目符号为十进制数字项目符号、第二个嵌套列表项的项目符号为大写字母项目符号、第三个嵌套列表项的项目符号为大写罗马数字项目符号，CSS 代码如下：

```
    OL { list-style: decimal }
    OL OL { list-style: upper-alpha }
    OL OL OL { list-style: upper-roman }
```

9.4.4 运行结果

在 IE 浏览器中运行 9-4.html 页面，其结果如图 9-11 所示。



图 9-11　文章详解页面

9.4.5 实例分析



源码解析：

在上面的实例中定义了三个嵌套列表，第一个嵌套列表的列表项的项目符号为 1、2、3…，在它的第二个列表项下面又嵌套了两个列表：第一个列表项的项目符号为 A、B、C…；第二个列表项的项目符号为 I、II、III…，如上面所述的这种格式可以使用 list-style 属性设置，代码简洁，使用灵活。

9.5 常见问题解答

9.5.1 list-style 和 list-style-type 在 IE 中不显示的问题



list-style 和 list-style-type 在 IE 中不显示的问题

网络课堂：http://bbs.itzcn.com/thread-12272-1-1.html

网页排版时会遇到“list-style”和“list-style-type”在 IE 中不显示，而在 Firefox 中正常显示的问题，这是怎么回事啊？

【解决办法】经过我的研究，发现<ul>元素在 IE 和 Firefox 中有“跟随且自动缩进”的特

性，就是当一个<ul>前面有<img>元素且设置了<img>的 align=“left”时，在 IE 和 Firefox 中的“list-style”和“list-style-type”都显示不正常，比如下面的代码：

```
<img src="684.jpg"  align="left"/>
<ul>
  <li>第一项 </li>
  <li>第二项</li>
  <li>第三项 </li>
</ul>
```

IE 的解决办法就是设置<ul>标签的“width”属性和“padding-left”属性，“padding-left”的值应该≥15，即：

```
<style type="text/css">
    ul{
         padding-left:15px;
         width:450px;
    }
</style>
```

或者使用<div>标签将<ul>包住，设置 div 元素的样式为：

```
float:left;
padding-left:15;
```

9.5.2　list-style-type 在 IE 与 Firefox 中奇怪的现象



list-style-type 在 IE 与 Firefox 中奇怪的现象
网络课堂：http://bbs.itzcn.com/thread-12275-1-1.html

最近在学习 CSS，就做了一个有关列表的页面，这是我的代码片段：

```
<style type="text/css">
.blockcont4 {
    height:277px;
    border: 1px #becbcb solid;
}
.olleft {
    margin:6px 16px 6px 38px;
    list-style-type:circle;
    width:auto;
}
.olleft li {
    height:22px;line-height:22px;white-space:nowrap;overflow:hidden;
    }
.olleft li a:link {
    display: block;height:22px;
    }
```

```
.olleft li a:hover {
    background-color: red;
    }
</style>
<body>
<div class="blockcont4">
<ol class="olleft">
<li>Div CSS 布局关于分辨率与 100%自适应问题的探讨！ </li>
<li>CSS 基础实例：CSS 实现带背景图片的文字链接的方法 </li>
<li>DIV CSS 实例教程：一款清新风格的 CSS 新闻列表制作  </li>
<li>CSS 布局教程：如何用 CSS 构建图片、文字混排的产品展示？ </li>
<li>CSS 布局教程：用 DIV CSS 实现国内经典式三行两列布局 </li>
<li><a>DIV CSS 实例教程：蔚蓝色的海洋 漂亮的 CSS 纵向菜单</a> </li>
<li>DIV CSS 菜单实例：红与黑 具有立体质感的 CSS 菜单 </li>
<li>DIV CSS 横向菜单实例：简单的思路 陷下的风格 不错的效果 </li>
</ol>
</div>
</body>
```

不知道为什么，上面这段代码的列表样式(list-style-type:circle;)在 Firefox 中不显示，但在 IE 中显示正常。在上面的代码中，我定义了超链接的 display:block，在 Firefox 下显示正常，突出显示部分(background-color:red;)，但在 IE 下却不显示，这是怎么回事啊？

【解决办法】对于上面提出的第一个问题，是因为没有定义 list-style-position，所以在 Firefox 中列表预设标记无法显示；第二个问题是由于没有定义 A 的宽度，故没有显示。需要修改上面 CSS 样式表中的两个类样式，修改内容如下：

```
.olleft {
    margin:6px 16px 6px 38px;
    list-style-type:circle;
    list-style-position: inside ;
    width:auto;
}
.olleft li a:link {
    display: block;height:22px;
    width:500px;
}
```

9.5.3　关于 ul ul ul 的 CSS 规则定义问题



关于 ul ul ul 的 CSS 规则定义问题
网络课堂：http://bbs.itzcn.com/thread-12277-1-1.html

在 Dreamweaver 8 的 CSS 定义中，可以有此类定义：ul ul 定义和 ul ul ul 定义。它们的 CSS 代码分别是 ul ul {list-style-type: disc}和 ul ul ul {list-style-type:square}。我只知道无序列表 ul 的 CSS 定义，请问：怎样理解 ul ul 或 ul ul ul 的 CSS 规则定义？

【解决办法】首先你要理解 CSS 命名规则，比如：

```
<div class="div1"><ul></ul></div>
```

如果你想定义这个 ul 的样式，可以直接这样定义：

```
ul{list-style-type;disc;}
```

但是这样定义太广泛了，这个是让所有的列表项的项目符号都为实心圆点。如果这个页面还有其他 ul，也会受到影响。为了精确地定义这个 div 下的 ul，可以这样定义：

```
.a1 ul{list-style-type:disc;}
```

代码表示类 a1 下的列表项的项目符号为实心圆点，这样不会影响到其他的列表样式。可以看出命名规则是这样的：在需要定义的样式前加上父级的类名，来精确地指定要给哪个列表定义样式。可以在前面加上无数个父级类名来区分要设置的样式。那么根据代码“ul ul{list-style-type:disc}”，意思就是设置 ul 下的 ul 的列表项的项目符号为实心圆点；而另一个“ul ul ul{list-style-type:square}”的意思就是设置 ul 下的 ul 下的 ul 的列表项的项目符号为实心方形项目符号。

9.6 习　　题

1. 填空题

(1) CSS 中可以使用__________属性为列表项定义项目符号。

(2) 在下面的代码中，如果在该页面所在的目录下不存在 p9.gif 图片时，“ol”列表项的项目符号为__________。

```
<ol style="list-style-image:url(p9.gif); list-style-type:circle;">
    <li>《世界博物馆巡礼系列》</li>
       <li>《老照片》</li>
       <li>《中华传统文化经典文库》</li>
</ol>
```

(3) list-style-position 属性的有效值有三个，分别是 inside、outside 和 inherit，默认值为__________。

(4) 在设置 list-style 属性时，如果既定义了 list-style-type 属性，又定义了 list-style-image 属性，那么无论哪个属性在前，将会以__________属性定义的列表样式出现。

2. 选择题

(1) 有这样的一段代码：

```
<ol style="list-style-type:decimal;">
        <li>《世界博物馆巡礼系列》</li>
        <li style="list-style-type:none">《大地玫瑰包系列》</li>
        <li style="list-style-type:none">《经济学家茶座》</li>
        <li>《鼠疫》</li>
```

```
        <li>《我把青春献给您》</li>
    </ol>
```

请问：第 4 个列表项的项目符号为________。

A. 2　　B. 3　　C. 4　　D. 5

(2) 在 CSS 中，可以使用________属性来实现图像列表。

A. list-style-image　　B. list-style-type

C. list-style-position　　D. list-style

(3) 怎样给所有的<h1>标签添加背景颜色？________

A. h1{background-color:#FFFFFF}

B. h1{background-color:#FFFFFF;}

C. h1all {background-color:#FFFFFF}

D. #h1{background-color:#FFFFFF}

(4) 如何能够定义列表的项目符号为实心方形？________

A. list-type:square　　B. type:2

C. type: square　　D. list-style-type:square

(5) list-style 属性是一种同时设置标签及其位置的简略方式，它的值包括 list-style-type、list-style-image 和 list-style-position 属性的所有合法值，可以按________顺序来设置属性值。

A. list-style-type list-style-image list-style-position

B. list-style-type list-style-position list-style-image

C. list-style-position list-style-image list-style-type

D. list-style-position list-style-type list-style-image

(6) 下面的 CSS 代码定义了两个列表，第一个列表的列表项的项目符号为________，而第二个列表的列表项的项目符号为________。

```
ol{
    list-style-type:circle;
    list-style-image:url(p9.jpg);
    list-style-position:outside;
}
ol{
    list-style-image:url(p9.jpg);
    list-style-type:circle;
    list-style-position:outside;
}
```

A. 实心圆点 p9.jpg 图片　　B. 都是实心圆点

C. 都是 p9.jpg 图片　　D. 都没有项目符号

3. 上机练习

上机练习：设计文档结构图效果页面

以某章的目录为例，设计文档结构图效果页面。

首先定义有序列表，列表项有 4 个，分别是滤镜特效、列表、定位与布局和鼠标指针。其中，需要在“滤镜特效”列表项下面定义一个嵌套的有序列表，列表项为：Alpha 滤镜、BlendTrans

滤镜、Blur 滤镜、Chroma 滤镜和 DropShadow 滤镜；还需要在“定位与布局”列表项下面定义嵌套列表，列表项包括：定位属性、边偏移属性、position 属性和 overflow 属性。运行结果如图 9-12 所示。



图 9-12　文档结构图页面的运行效果



第 10 章　滚动条属性

内容摘要：

我们在网页中浏览内容时，当网页内容显示超出浏览器可视范围时，就会出现滚动条。其实，在一个层中也可以出现滚动条。要在层中出现滚动条，我们可以使用 CSS 设置层的滚动条属性，而且还可以设置滚动条边的颜色等。在本章中将详细介绍如何使用 CSS 的滚动条属性。

学习目标：

- 掌握如何设置滚动条亮边框颜色
- 掌握如何设置滚动条 3D 界面亮边框颜色
- 掌握如何设置滚动条 3D 表面的颜色
- 掌握如何设置滚动条方向箭头的颜色
- 掌握如何设置滚动条 3D 界面的暗边颜色
- 掌握如何设置滚动条暗边框颜色
- 掌握如何设置滚动条基准颜色
- 掌握如何设置滚动条拖动区域颜色

10.1　制作带有彩色 3D 亮边框的滚动条

在网页中，我们常常会看到滚动条，但这些经浏览器支持的滚动条，大多形式比较单一，CSS 的样式可以改变这种单一的形式，例如给滚动条的亮边框添加颜色等。

**视频教学：**光盘/视频/10/10.1　制作带有彩色 3D 亮边框的滚动条.avi　长度：5 分钟

10.1.1　基础知识——scrollbar-3dlight-color

在一般页面中，滚动条都显示在浏览器的右侧。而滚动条 3D 亮边框是指右侧滚动条中显示在滑块和箭头区域最外层左侧和顶部的区域。在我们使用之前，先看看设置滚动条亮边框颜色的语法结构。

滚动条的 3D 亮边框颜色属性语法结构如下所示：

```
scrollbar-3dlight-color: color;
```

一般情况下我们会先在页面中定义一个 div，当在 div 内输入的内容过多时会给这个 div 定义一个滚动条属性，为这个滚动条添加一些修饰属性无疑会给页面增加亮点。

例如，下面的实例代码：

```
div{
overflow:scroll;
scrollbar-3dlight-color:#000000;
}
```

在 CSS 的 div 样式里添加这些代码，就可添加滚动条属性，并为滚动条亮边框设置颜色为 #000000。



当在<html>元素中定义了滚动条 3D 亮边框颜色属性后，其子元素都将继承这个属性中定义的显示效果。

10.1.2　实例描述

有时一张页面不足以显示全部的内容，这时浏览器会显示自带的滚动条，而这些滚动条在浏览器中都有固定的样式。但是可以使用 CSS 的样式给我们自定义的 div 的滚动条添加样式，比如为滚动条设置亮边框的颜色。

10.1.3　实例应用

【**例 10-1**】制作带有彩色 3D 亮边框的滚动条

(1) 新建一个 HTML 页面，并进行布局，保存为 index.html。

(2) 接下来在 div 标记 box_greyc 里写入以下代码。

```
<div class= box_grey >
<strong>蓝色心情许愿墙</strong></h3>
                <p> 开夜的天窗，对着划过的流星许下永恒的祝愿，不变的诺言。在每个人的心底，
都有最美好的祝愿。或许，这就是一种缘份。<br />
<br />
                让我们用最虔诚的心，在这里许下您的心愿、祝福，留下您最想对他(她)说的话，
永久地挂在许愿墙上，愿望就会很快实现的哟。<br />
                </p></div>
```

(3) 下面来对这个 div 定义样式，首先定义这个 div 的高度、宽度和边框属性，这里我们要用到滚动条，所以就要定义滚动条及设置滚动条亮边框的颜色。代码如下：

```
div .box_grey{
      height:200px;
      overflow:scroll;
      scrollbar-3dlight-color:#F00;
    }
```

10.1.4　运行结果

在浏览器中直接打开实例 HTML 文件。此时会看到如图 10-1 所示的页面，在这个页面中我们可以看到滚动条中红色的部分就是要设置的亮边框的颜色。



图 10-1　设置彩色 3D 亮边框的实例运行效果

10.1.5　实例分析

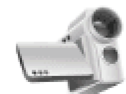


源码解析：

在这个实例中，我们用到了设置滚动条亮边框颜色属性来实现亮边框的效果，学会之后，我们可以随意地设置想要的颜色。

10.2　制作带有彩色亮边框的滚动条

滚动条亮边框是指右侧显示的滚动条中与滚动条 3D 亮边框相邻的区域。使用 scrollbar-highlight-color 属性可以设置滚动条 3D 界面亮边框颜色。

 **视频教学：**光盘/视频/10/10.2　制作带有彩色亮边框的滚动条.avi　　长度：4 分钟

10.2.1　基础知识——scrollbar-highlight-color

scrollbar-highlight-color 属性可以设置滚动条 3D 界面亮边框颜色。在使用时要注意，在单独使用滚动条亮边框颜色属性时，将对滚动条其他部分的颜色产生影响。

语法格式如下：

```
scrollbar-highlight-color: color;
```

亮边框是滚动条中和 3D 亮边框相邻的部分，我们可以为它设置不同的颜色，以达到为页面增色的效果。

例如以下实例代码：

```
div{
overflow:scroll;
scrollbar-highlight-color:#F00000;
}
```

10.2.2　实例描述

通常浏览器的滚动条中的亮边框有固定的颜色，我们在网页中使用滚动条时，会出现与要设计的网页颜色不搭配的现象，这时我们就可以通过设置滚动条亮边框的颜色来对页面颜色进行协调。

10.2.3　实例应用

【例 10-2】制作带有彩色亮边框的滚动条

(1) 新建一个 HTML 页面，设置背景颜色为#FC9，CSS 样式代码如下：

```
body{
    padding:10px 0 0 0;
    background-attachment: fixed;
    background-color: #FC9;
    padding-left:100px;
}
```

(2) 在页面中新建一个 div 标记，在里面写入一段文字，以便出现滚动条。

```
<div align="center">朋友如海，宽容作舟，泛舟于海，方知海之宽阔；朋友如山，宽容为径，循径登山，方知山之高大；朋友交心，交而知心，心心相印，方知心之高尚。” <br />
  人的一生难得有几个知心朋友，也许有人说那也不是啊，我就有好多的知心朋友。可是或许你对他们是真心的，而他们是不是和你对他们一样就不得而知了。我见过好多表面上好得跟一个人似的，但是私下里却是你说我的不是我说你的不对，我觉得这样还不如不做朋友的好。 <br />
  我觉得自己还算是比较幸运的了，在学校时有谈得来的朋友。参加工作了更让我遇到一个我特别感激的朋友，虽然说我俩的性格相差十万八千里。她属于很活泼，很健谈的那种，而我却是性格内向，不善言谈的。可是只要我们俩到一起就有说不完的话，说给其他同事听他们还都不信，因为他们都知道我的性格，是那种可能别人说十句，我才回一句的那种人，可能也因此让他们觉得我不好接近，可是实际上我内心不是这样想的，可是就是跟他们在一起就觉得紧张说不出话来，也不知道是什么原因。说实话，连我自己都觉得奇怪，我一般和不太熟悉的人话上特别少的，总觉得跟他们没有什么话可以延伸下去，可是跟她第一次见面我们就聊了一个晚上，真的特别开心能有这样一个朋友，有什么话都可以跟她说。而在平时她就像一个大姐姐一样。我的依赖思想特别重，做什么事总想着前面有人带着，所以每次有什么事她都在前面我跟在后面做，当然我也知道这个不好所以我尽量改正自己的这个缺点。我是一个情感不外露的人，不管我心里对她是多么的感激，我都不会把感谢常挂在嘴上，倒不是因为别的什么而是我觉得这样的话放在心里就可以了不需要常常拿出来说。 <br />
  知心的朋友真的是很难得的，如果你拥有了这份友谊就不要轻易地去破坏它。 <br />
</div>
```

(3) 接下来定义 div 的 CSS 样式，如宽度、高度、边框大小和滚动条属性等。代码如下：

```
div
{
    height:300px;
    width:400px;
    padding:10px 0 0 0;
    border:1px solid #666666;
    overflow:scroll;
    scrollbar-highlight-color:#F00000;
}
```

(4) 保存 HTML 页面，完成 CSS 样式的定义。

10.2.4　运行结果

将页面在浏览器中打开，我们可以看到如图 10-2 所示滚动条的亮边框的颜色为橘红色。



图 10-2　设置滚动条亮边框颜色的实例运行效果

10.2.5 实例分析



源码解析：

在以上实例中，我们使用了滚动条属性 scrollbar-highlight-color 来设置滚动条亮边框的颜色。以后可以使用这个属性来为我们的网页设计出绚丽的滚动条。



在单独使用滚动条亮边框颜色属性时，将对滚动条其他部分的颜色产生影响。

10.3 制作彩色滚动条区块

滚动条 3D 表面也就是滚动条的滑块的表面，指右侧显示的滚动条中，可以使用鼠标上下拖动的区域，以及箭头区域部分。我们可以通过滚动条的 scrollbar-face-color 属性来设置它表面的颜色。

 **视频教学：**光盘/视频/10/10.3　制作彩色滚动条区块.avi　　长度：4 分钟

10.3.1 基础知识——scrollbar-face-color

scrollbar-face-color 属性是用来设置滚动条 3D 表面的颜色的，在设置颜色时，要注意页面之间的颜色搭配。

语法格式如下：

```
scrollbar-face-color color;
```

例如我们可以通过上面的语法在 CSS 样式中给 div 层中的滚动条的表面设置颜色，如设置红色，代码如下：

```
div
{
 height:100px;
 width:400px;
 padding:80px 0 0 0;
 border:1px solid #666666;
 overflow:scroll;
 scrollbar-face-color:#F00000;
}
```

上述代码中，先使用 overflow 属性为 div 添加滚动条，然后使用 scrollbar-face-color 属性添加滚动条 3D 表面的颜色为红色(#F000000)。



设置滚动条表面颜色时，应注意颜色的搭配。必须只有在滚动条出现的情况下，表面的颜色效果才会显示。

10.3.2　实例描述

在网页中我们需要在一定的区域放置很多不同的内容，但毕竟一张页面内容承载量是有限的。我们可以通过增加滚动条，在一个页面中添加更多的内容，有时这些内容的区块需要在页面中突出显示，我们可以通过设置滚动条的 3D 表面的不同颜色来区分各个不同的区块。

10.3.3　实例应用

【例 10-3】彩色滚动条区块

(1) 新建一个 HTML 页面，并进行布局，保存为 index.html。

(2) 接下来在 div 标记 box_greyc 中写入以下代码。

```
<div class= box_grey >
<strong>蓝色心情许愿墙</strong></h3>
                <p> 开夜的天窗，对着划过的流星许下永恒的祝愿，不变的诺言。在每个人的心底，
都有最美好的祝愿。或许，这就是一种缘份。<br />
<br />
                让我们用最虔诚的心，在这里许下您的心愿、祝福，留下您最想对他(她)说的话，
永久地挂在许愿墙上，愿望就会很快实现的哟。<br />
                </p></div>
```

(3) 定义 div2 的 CSS 样式：我们把滚动条表面颜色定义为蓝色，其代码如下：

```
div .box_grey{
      height:200px;
      overflow:scroll;
     scrollbar-face-color:#F00;
    }
```

(4) 保存这个页面，命名为 index.html，这样就把 CSS 样式定义好了。

10.3.4　运行结果

将页面 index.html 在浏览器中运行，可以看到效果如图 10-3 所示。



图 10-3　设置彩色滚动条区块的实例运行效果

10.3.5　实例分析



源码解析：

以上实例是通过设置滚动条的颜色属性 scrollbar-face-color 来实现的，通过这个属性，我们可以把滚动条分成颜色不同的区块。



在设置这种不同的区块时，由于我们用到的是滚动条，所以要把滚动条设置成始终显示。

10.4　制作突出方向箭头

滚动条箭头是指右侧滚动条中在滚动条的顶部和底部显示的三角箭头区域部分，单击三角区域可以使滚动条上下滚动。我们可以使用 scrollbar-arrow-color 属性来为滚动条方向箭头设置颜色。



视频教学：光盘/视频/10/10.4　制作突出方向箭头.avi　　　　**长度：**3 分钟

10.4.1　基础知识——scrollbar-arrow-color

scrollbar-arrow-color 属性可以在 CSS 样式中设置滚动条方向箭头的颜色。方向箭头一般有默认的颜色。我们通过这个属性来设置方向箭头不同的颜色，可以使箭头在网页中起到突出显示的效果。

使用语法格式如下：

```
scrollbar-arrow-color: color
```

例如我们可以通过上面的语法在 CSS 样式中给 div 层中的滚动条的方向箭头设置颜色，如设置成红色，代码如下：

```
div
{
 scrollbar-arrow-color:#F00000;
 height:100px;
 width:400px;
 padding:100px 0 0 0;
 border:1px solid #666666;
 overflow:scroll;
}
```

上述代码中，我们先使用 overflow 属性为 div 添加滚动条。然后使用 scrollbar-arrow-color 属性设置滚动条方向箭头的颜色为红色(#F000000)。



在设置时注意，颜色尽量不要与页面的颜色相同。

10.4.2　实例描述

有时在使用滚动条时，虽然用鼠标滚动页面很方便，但是一些细小的滚动还是需要借助鼠标左键单击滚动条的方向箭头来完成。可是有时这些箭头与页面的颜色很接近，要找到它并单击可能不太方便。这时我们就可以使用滚动条的 scrollbar-arrow-color 属性来为其设置突出的颜色，以方便快速地查找到滚动条的方向箭头。

10.4.3　实例应用

【例 10-4】制作突出方向箭头

(1) 新建一个 HTML 页面，并进行布局，保存为 index.html。

(2) 接下来在 div 标记 box_greyc 里写入以下代码：

```
<div class= box_grey >
<strong>蓝色心情许愿墙</strong></h3>
                <p> 开夜的天窗，对着划过的流星许下永恒的祝愿，不变的诺言。在每个人的心底，都有最美好的祝愿。或许，这就是一种缘份。<br />
<br />
                让我们用最虔诚的心，在这里许下您的心愿、祝福，留下您最想对他(她)说的话，永久地挂在许愿墙上，愿望就会很快实现的呦。<br />
                </p></div>
```

(3) 定义 div 的 CSS 样式：包括高度、宽度、边框的粗度和颜色、滚动条，并使用 scrollbar-arrow-color 属性来设置滚动条方向箭头的颜色为红色，代码如下所示：

```
div .box_grey{
 scrollbar-arrow-color:#FF0000;
```

```
    height:200px;
        overflow:scroll;
      scrollbar-face-color:#F00;
}
```

10.4.4 运行结果

将保存的页面 index.html 在 IE 浏览器中运行，可以看到如图 10-4 所示的效果，有 scrollbar-arrow-color 属性定义的红色方向箭头。



图 10-4　设置红色突出方向箭头的实例运行效果

10.4.5 实例分析



源码解析：

在本实例中，看到的效果是通过滚动条颜色属性的 scrollbar-arrow-color 属性来实现的，我们可以看到图 10-4 所示方框中滚动条的箭头为红色，而且我们还可以为它设置不同的颜色。

10.5 制作突出滚动条阴影效果

滚动条 3D 界面的暗边也就是滚动条的阴影部分，主要是指滚动条中显示在滚动条拖动区域和箭头区域的右侧和底侧部分。我们可以使用滚动条的 scrollbar-shadow-color 属性来为它设置颜色。



视频教学：光盘/视频/10/10.5　制作突出滚动条阴影效果.avi　　长度：4 分钟

10.5.1　基础知识——scrollbar-shadow-color

scrollbar-shadow-color 属性是 CSS 中滚动条的颜色设置属性，我们可以在 CSS 的样式中设置滚动条 3D 界面的暗边--也就是滚动条阴影的颜色。

语法格式如下：

```
scrollbar-shadow-color:color;
```

这样便可以通过上面的语法在 CSS 样式中给滚动条的 3D 界面的暗边设置颜色，在这里设置的颜色为蓝色(#0000FF)，其代码如下：

```
div
{
overflow:scroll;
 scrollbar-shadow-color:#0000FF;
 height:100px;
 width:100px;
 border:1px solid #666666;
}
```

在以上代码中，首先用 overflow 属性设置层中的滚动条始终显示，然后用 scrollbar-shadow-color 设置滚动条 3D 界面的暗边的颜色为蓝色。



滚动条 3D 界面的暗边的颜色最好不要与暗边的颜色一致或相近，不然效果会不太明显。

10.5.2　实例描述

在设计页面时，好的阴影效果会给页面增加不少亮点，如果阴影的颜色设计得当，更是锦上添花。在设置滚动条时，通常会用到滚动条的阴影效果，也就是滚动条的 3D 界面的暗边，我们可以使用 CSS 样式中滚动条的 scrollbar-shadow-color 属性设置适合的颜色，以突出页面的效果。

10.5.3　实例应用

【例 10-5】突出滚动条阴影效果

(1) 新建一个 HTML 页面，并进行布局，保存为 index.html。

(2) 接下来在 div 标记 box_greyc 中写入以下代码：

```
<div class= box grey >
<strong>蓝色心情许愿墙</strong></h3>
```

```
                <p> 开夜的天窗，对着划过的流星许下永恒的祝愿，不变的诺言。在每个人的心底，
都有最美好的祝愿。或许，这就是一种缘份。<br />
<br />
                让我们用最虔诚的心，在这里许下您的心愿、祝福，留下您最想对他(她)说的话，
永久地挂在许愿墙上，愿望就会很快实现的哟。<br />
                </p></div>
```

(3) 定义 div 的 CSS 样式：包括 div 的高度、宽度、边框的粗度和颜色，其代码如下：

```
div{
height:100px;
 width:400px;
 padding:100px 0 0 0;
 border:1px solid #666666;
}
```

(4) 把用于定义滚动条的 3D 界面的暗边颜色的 CSS 属性添加到步骤(3)的代码中，这里我们通过 overflow 属性设置滚动条始终显示，其代码如下：

```
div .box_grey
{
height:200px;
overflow:scroll;
scrollbar-shadow-color:#FF0000;
}
```

10.5.4　运行结果

将保存的名为 index.html 的页面在 IE 浏览器中运行，我们可以看到如图 10-5 所示的红色滚动条 3D 界面的暗边效果。



图 10-5　突出的滚动条阴影效果

10.5.5　实例分析



源码解析：

在本实例中我们使用 scrollbar-shadow-color 属性来设置滚动条 3D 界面的暗边颜色。

10.6　制作彩色滚动条暗边框

滚动条暗边框也就是滚动条 3D 界面暗边的阴影部分，指右侧滚动条中显示在滚动条隐隐相邻的左侧和顶部区域。我们可以使用 CSS 样式中的滚动条颜色属性 scrollbar-darkshadow-color 来设置它的颜色。

 **视频教学：**光盘/视频/10/10.6　制作彩色滚动条暗边框.avi　　长度：4 分钟

10.6.1　基础知识——scrollbar-darkshadow-color

scrollbar-darkshadow-color 是 CSS 样式中设置滚动条暗边颜色的属性。通过这个属性可以在页面中的滚动条中设置滚动条暗边的颜色。

语法格式如下：

```
scrollbar-darkshadow-color:color;
```

我们通过一个实例，来具体地学习这个语法的使用，其代码如下：

```
div
{
height:100px;
 width:400px;
overflow:scroll;
 scrollbar-darkshadow-color:#00F;
}
```

在以上代码中，首先使用 overflow 属性设置在 div 中始终显示滚动条，然后通过 scrollbar-darkshadow-color 属性来设置滚动条暗边颜色的 CSS 样式为蓝色(#00F)



滚动条暗边，实际上就是 3D 滚动条的阴影部分的颜色。请大家不要将其与 3D 界面的暗边相混淆。

10.6.2　实例描述

在设计页面时，有时会需要滚动条，但滚动条中的 3D 界面的暗边，常常会被我们忽视，而和暗边非常相近的暗边框则更是不会引起注意。但在网页中，确实有时需要用到滚动条暗边

框，我们可以使用 CSS 样式中滚动条的 scrollbar-darkshadow-color 属性来设置适合的颜色，以突出页面的效果。

10.6.3　实例应用

【例 10-6】制作彩色滚动条暗边框

(1) 新建一个 HTML 页面，并进行布局，保存为 index.html。

(2) 接下来在 div 标记 box_greyc 中写入以下代码：

```
<div class= box_grey >
<strong>蓝色心情许愿墙</strong></h3>
                <p> 开夜的天窗，对着划过的流星许下永恒的祝愿，不变的诺言。在每个人的心底，
都有最美好的祝愿。或许，这就是一种缘份。<br />
<br />
                让我们用最虔诚的心，在这里许下您的心愿、祝福，留下您最想对他(她)说的话，
永久地挂在许愿墙上，愿望就会很快实现的哟。<br />
                </p></div>
```

(3) 定义 div 的 CSS 样式：包括 div 的高度、滚动条暗边颜色等，其代码如下：

```
div .box_grey
{
height:200px;
overflow:scroll;
scrollbar-darkshadow-color:#F00;
}
```

10.6.4　运行结果

在 IE 浏览器中运行名为 index.html 的网页，可以看到如图 10-6 所示中蓝色方框的效果。



图 10-6　设置暗边框颜色的实例运行效果

10.6.5　实例分析



源码解析：

在本实例中我们使用 scrollbar-darkshadow-color 属性来设置滚动条的暗边框颜色。

10.7　制作滚动条基准色

网页中的滚动条是由 3D 界面亮边框、3D 表面、3D 界面的暗边、方向箭头、暗边框和拖动区域组成的。在 CSS 样式中，每一个部分都有一种颜色属性设置。我们还可以通过一种属性来设置它们的基准颜色，然后在此基础上增加额外的颜色设置，这种属性就是滚动条基准颜色属性 scrollbar-base-color。

 **视频教学：**光盘/视频/10/10.7　制作滚动条基准色.avi　　长度：5 分钟

10.7.1　基础知识——scrollbar-base-color

scrollbar-base-color 属性是设置滚动条的基准颜色。这种基准颜色是滚动条的背景颜色，我们还可以在此基础上使用其他属性。

使用语法如下：

```
scrollbar-base-color:color;
```

例如，在层中使用 scrollbar-base-color 属性设置滚动条基准颜色为蓝色，其 CSS 样式代码如下：

```
div{
    overflow:scroll;
    scrollbar-base-color: #F00;
    height: 100px;
    width: 200px;
    }
```



滚动条的基准颜色只是滚动条的基本颜色，在使用 scrollbar-base-color 属性设置的同时，其他的滚动条颜色设置属性可以同时使用。

10.7.2　实例描述

在设计页面时，我们经常会在页面中设置其背景颜色，也就是通常所说的基准颜色。在此基础上进行的页面设计，会使页面更加美观。而我们在使用滚动条时也会用到这种基准颜色的

设置。下面我们来做一个设置滚动条基准颜色的实例吧。

10.7.3 实例应用

【例 10-7】制作滚动条基准色

(1) 新建一个 HTML 页面，并进行布局，保存为 index.html。

(2) 接下来在 div 标记 box_greyc 中写入以下代码：

```
<div class= box_grey >
<strong>蓝色心情许愿墙</strong></h3>
                <p> 开夜的天窗，对着划过的流星许下永恒的祝愿，不变的诺言。在每个人的心底，
都有最美好的祝愿。或许，这就是一种缘份。<br />
<br />
                让我们用最虔诚的心，在这里许下您的心愿、祝福，留下您最想对他(她)说的话，
永久地挂在许愿墙上，愿望就会很快实现的哟。<br />
                </p></div>
```

(3) 定义 div 的 CSS 样式：包括 div 的高度、滚动条的基准颜色样式为红色、滚动条 3D 表面颜色样式为灰色等，在这里使用 overflow 设置滚动条始终显示，其代码如下：

```
div .box_grey{
      height:200px;
      overflow:scroll;
      scrollbar-face-color:#CCC;
      scrollbar-base-color: #F00;
   }
```

10.7.4 运行结果

在 IE 浏览器中运行名为 index.html 的页面，可以看到如图 10-7 所示的效果，其中滚动条的基准颜色为红色。



图 10-7　设置滚动条基准颜色的实例运行效果

10.7.5　实例分析



源码解析：

在本实例中，我们使用 scrollbar-base-color 属性将滚动条的基准颜色设置为红色，然后又设置滚动条 3D 表面的颜色为灰色，由此可见滚动条的颜色属性可以同时使用。

10.8　制作彩色滚动条拖动区域

滚动条拖动区是指右侧滚动条中可以使用鼠标进行上下拖动的区域的背景部分。在这部分中，我们可以通过 CSS 样式中的滚动条颜色属性 scrollbar-track-color 来对其颜色进行修饰。



视频教学：光盘/视频/10/10.8　制作彩色滚动条拖动区域.avi　　　长度：4 分钟

10.8.1　基础知识——scrollbar-track-color

scrollbar-track-color 是 CSS 样式中滚动条的颜色设置属性，不过这个属性只用来设置滚动条拖动区域的颜色，即滚动条滑块下的区域的颜色。

语法格式如下：

```
scrollbar-track-color:color;
```

在使用的时候，我们可以在 CSS 样式中使用这个属性，例如把滚动条的拖动区域的颜色设置为蓝色，其代码如下：

```
div{
   overflow: scroll;
   scrollbar-track-color: #00F;
}
```

在以上代码中，我们首先使用 overflow 属性设置在 div 里始终显示滚动条，然后通过 scrollbar-track-color 属性来设置滚动条暗边颜色的 CSS 样式为蓝色(#00F)。



scrollbar-track-color 属性的设置不仅仅限于 div 中，在 HTML 标记中使用该属性可以使整个页面的滚动条的拖动区域颜色一致。

10.8.2　实例描述

大家都以为滚动条在页面中存在是最平常的事情了，但在一些高质量的页面中，如含有图片较多的页面中使用的就更频繁。为了达到和页面很好地契合，我们往往会在滚动条上添加一些修饰，如改变拖动区域的颜色。要改变拖动区域的颜色，可以使用 CSS 样式中滚动条的

scrollbar-track-color 来设置，下面就让我们实现一个实例，来看看效果吧。

10.8.3 实例应用

【例 10-8】制作彩色滚动条拖动区域

(1) 新建一个 HTML 页面，并进行布局，保存为 index.html。

(2) 接下来在 div 标记 box_greyc 中写入如下代码：

```
<div class= box_grey >
<strong>蓝色心情许愿墙</strong></h3>
               <p> 开夜的天窗，对着划过的流星许下永恒的祝愿，不变的诺言。在每个人的心底，
都有最美好的祝愿。或许，这就是一种缘份。<br />
<br />
               让我们用最虔诚的心，在这里许下您的心愿、祝福，留下您最想对他(她)说的话，
永久地挂在许愿墙上，愿望就会很快实现的呦。<br />
               </p></div>
```

(3) 定义 div 的 CSS 样式：包括高度、滚动条的拖动区域的颜色等，这里我们使用 overflow 属性定义层中的滚动条始终显示，其代码如下：

```
div .box_grey{
      height:200px;
      overflow:scroll;
      scrollbar-face-color:#CCC;
     scrollbar-track-color:#F00;
   }
```

10.8.4 运行结果

将名为 index.html 的页面在 IE 浏览器中运行，可以看到如图 10-8 所示的效果，其中的红色部分就是滚动条拖动区域的颜色。



图 10-8 设置拖动区域颜色的实例运行效果

10.8.5　实例分析



源码解析：

在本实例中，我们使用滚动条颜色属性 scrollbar-track-color 设置了滚动条拖动区域的颜色，这里只是使用了基本设置。根据需要我们还可以使用滚动条颜色属性的综合设置。

10.9　常见问题解答

10.9.1　滚动条颜色控制失效



在 XHTML 页面中滚动条颜色 CSS 控制失效问题。
网络课堂：http://bbs.itzcn.com/thread-12420-1-1.html

下面的代码为什么在 HTML 中能够显示，而在 XHTML 中会失效呢？

```
body{
 scrollbar-3dlight-color:#D4D0C8;        /*- 最外左 -*/
  scrollbar-highlight-color:#fff;        /*- 左二 -*/
  scrollbar-face-color:#E4E4E4;          /*- 面子 -*/
  scrollbar-arrow-color:#666;            /*- 箭头 -*/
  scrollbar-shadow-color:#808080;        /*- 右二 -*/
  scrollbar-darkshadow-color:#D7DCE0;    /*- 右一 -*/
  scrollbar-base-color:#D7DCE0;          /*- 基色 -*/
  scrollbar-track-color:#;               /*- 滑道 -*/
}
```

【解决办法】在 XHTML 中 body 不是根元素，只有 html 是根元素，而页面的滚动条也是属于根元素的，所以定义 body 没有效果的原因是我们只定义了一个子元素，把 body 换成*或者 xhtml 就可以了。

10.9.2　滚动条滑块的问题



CSS 中关于滚动条滑块的问题。
网络课堂：http://bbs.itzcn.com/thread-12421-1-1.html

各位，我就问个简单的滑块问题，我在 HTML 代码中设置 body 元素样式如下：

```
body{overflow:scroll;background-image:url('image.jpg');}
```

里面的那个 image.jpg 是一个比桌面大的图片。结果在浏览器中运行，滚动条是有了，而滑块没有。后来把样式应用到 body 中的一个 div 元素上，结果还是这个效果。是不是我有什么属性没设置？

【解决办法】首先 overflow:scoll 表达的意思是当页面内容超出一屏(或者给 body 设置的宽度)时，需要显示滚动条。当没有超出时，只显示 X 和 Y 的轴。问题在于背景图片不能设置大小。HTML 不会计算出背景图片的大小，这与直接插入的图片是有区别的。

我们分两步解决这个问题，第一步把 body 的宽度和高度与图片真实大小设置为一致，第二步给 div 设置的宽度和高度与图片一致即可。

10.10 习　　题

1. 填空题

(1) 在滚动条颜色属性中，使用__________设置滚动条 3D 表面的颜色。

(2) __________属性是用来改变滚动条拖动区域颜色的。

(3) __________属性是设置滚动条方向箭头的颜色的。

2. 选择题

(1) 要实现滚动条亮边框颜色效果，使用__________属性。

A. scrollbar-face-color　　B. scrollbar-darkshadow-color

C. scrollbar-highlight-color　　D. scrollbar-track-color

(2) 用__________属性实现滚动条 3D 界面的暗边颜色效果。

A. scrollbar-highlight-color　　B. scrollbar-track-color

C. scrollbar-shadow-color　　D. scrollbar-face-color

(3) 用__________属性实现设置滚动条的基准颜色效果。

A. scrollbar-track-color　　B. scrollbar-base-color

C. scrollbar-face-color　　D. scrollbar-darkshadow-color

3. 上机练习

上机练习：使用 Dreamweaver 创建一个简单的 HTML 页面，其中要用到 CSS 滚动条颜色属性。

通过本章的学习，我们已经掌握了 CSS 中滚动条颜色属性的使用方法，在本章的上机练习中，要求读者熟练使用这些属性，来实现如图 10-9 所示的彩色滚动条效果。



图 10-9　彩色滚动条的效果



第 11 章　统揽全局——页面的定位与布局

内容摘要：

在设计网页时，能否控制好各个模块在页面中的位置是非常关键的。

以前很多人喜欢用表格来控制页面的布局，因为这种方法简单、容易实现。但是它的缺点也不少：页面混乱、代码冗长、加载很慢、不易搜索引擎收录等。

后来，逐渐开始流行使用 div+CSS 来控制布局的方式。当然，使用 div+CSS 控制布局的方式克服了使用表格来控制布局的诸多弊端，所以成为多数人选择的页面布局方式。

本章首先介绍什么是盒子模型，然后讲解控制页面元素的定位方式、层叠顺序和元素位置，接着讲解如何使用 CSS 控制 div 对页面进行布局，最后讲解如何使用 CSS 控制页面元素制作一些特定效果。

学习目标：

- 了解什么是盒子模型
- 掌握元素定位的方法
- 掌握 div 布局的方法
- 掌握使用 CSS 样式显示一个页面元素的局部内容
- 掌握对页面容器内容溢出的处理方法
- 掌握页面元素的显示方式

11.1 盒子模型

在前面的章节中我们已经简单地提到介绍过盒子模型，下面我们来详细了解一下什么是盒子模型。

刚听到盒子模型(Box Model)的时候，每个人都会觉得很困惑。这的确是一个有点抽象的名词。

我们生活中的盒子，一般来说都是一个个的三维容器。最常见的就是文具盒、礼品盒、储物箱等。这些东西通常是一个具有长、宽、高等描述大小属性的立方体，我们可以用它来存放东西。

Web 页面中的“盒子”和现实中的“盒子”作用一样，都是为了存放东西，或者称之为容器。

现实世界是一个三维的世界，而电脑显示器为我们展示的是一个二维画面，无论它怎样去模仿三维效果(3D)，都不可能摆脱二维画面的本质。在 Web 页面中，更是如此。

所以我们不得不面对一个事实，Web 页面中的盒子将是一个平面的“盒子”。从这里或许就开始有点抽象了。不过我们先接受一个事实：平面“盒子”没有“高”这个属性，所以页面中的几乎所有对象对于大小属性的描述最多有长和宽两个方向的选项。而页面中的内容归根结底是为了显示给我们看的，我们人的眼睛接收到的都是图像信息，包括 Web 页面中的图像、文字、动画、视频等。



这些页面中的图像、文字、动画、视频才是我们需要为用户传递的信息，也就是 Web 页面最终要处理的对象。

这些 Web 页面中的图像、文字、动画、视频在计算机中都是以一个具有长、宽属性的方块呈现的，而页面布局就是为了更合理地摆放这些东西。所以我们要讲的“盒子”就是为了存放这些东西，并将这些东西呈现在页面中的适当位置。

举一个例子。假如你是一个仓库管理员，老板给你一大堆东西(假设有书籍等遇潮湿易损毁的东西，有玻璃制品等怕碰怕摔的东西……)，让你把这些东西放到整个仓库中去，而仓库是一个破旧地下室(我们知道一般地下室都比较容易受潮)。

这时我们就不得不考虑以下几个因素：为了易于搬运，我们需要选择足够厚度、足够结实的箱子存放；因为书籍等怕潮湿，我们必须保证其离墙壁有一段距离(当然通风是必需的)，玻璃制品怕碰，我们需要在箱子里垫上足够厚度的保护层。

整个规划中主要体现出了以下几点：盛装物品的“盒子”的厚度、“盒子”与墙壁(其他物体)的距离和盒子内部填充物(保护层)的厚度。

到此为止，我们知道了如何使用“盒子”存放物品。

同样，在 Web 页面中的“盒子”也是同样的结构：厚度、边距(边缘与其他物体的距离)、填充(填充厚度)。引申到 CSS 中，就是 border、margin 和 padding。当然，不能少了内容。也

就是说整个盒子在页面中占的位置大小，应该是内容的大小加上填充的厚度，再加上边框的厚度，还要加上它的边距。盒子模型如图 11-1 所示。

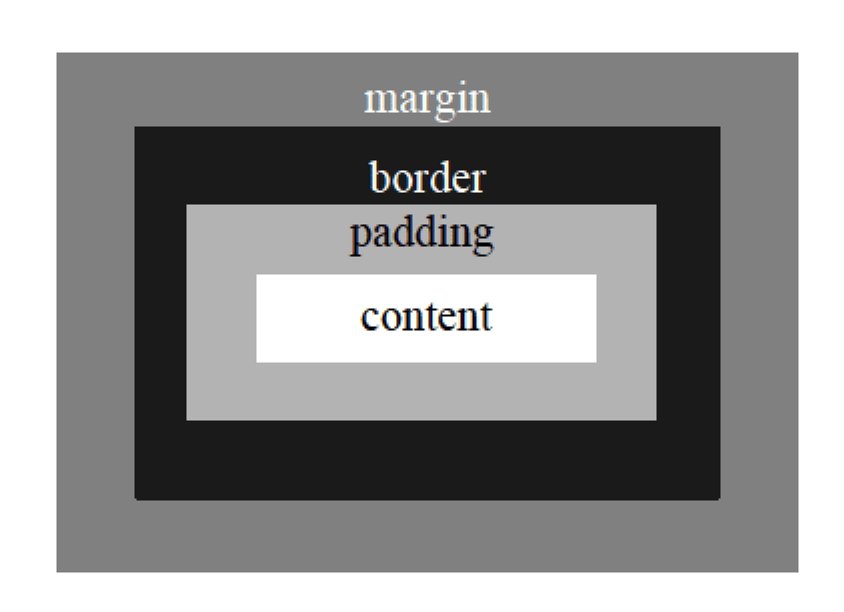


图 11-1　盒子模型示意图



我们需要牢记的是，现实生活中我们使用盒子是为了包装和保护物品，而在 Web 页面中使用盒子，完全是为了美观。所以理解“盒子模型”的时候一定要注意。

因为 HTML 页面是一个层级的“树”，所以盒子的嵌套使用是很正常的，就像平时我们使用盒子分门别类地存放东西，最后放到一个大箱子里一样。

总体来说，使用“盒子模型”这个名称还是非常形象的。它和现实中的盒子非常相似。如果你能把现实世界中的三维的盒子抽象到二维平面中，也就理解什么是“盒子模型”了。

11.2　具有固定位置的广告框

很多网站通常会在页面中放置一些广告框。如果将广告作为页面模块就会显得不够醒目，广告效果也不明显，所以很多站长喜欢把广告做成一些页面的侧边栏一直悬停在页面边上，这样既可以弥补页面的空白，也可以使广告更加醒目。

大家通常使用 JavaScript 控制广告模块的位置，这种方法有一种拖拽的延迟效果。

其实还可以使用 CSS 定位的方法，使广告框固定在浏览器窗口中的某一个位置。这样浏览器中页面滚动的时候广告似乎和窗口是浑然一体的，完全不会晃动。

**视频教学**：光盘/视频/11/ 11.2　具有固定位置的广告框.avi　　长度：6 分钟

11.2.1　基础知识——定位方式和位置

将页面容器定位在浏览器窗口中的固定位置需要设置两个 CSS 属性：定位方式 position 和相应的元素位置。

1. 定位方式 position 属性

position 属性是声明元素的定位方式，其语法如下：

```
position : static | absolute | fixed | relative
```

从其语法格式的定义可以知道，它有 4 个可选值，分别代表 4 种定位方式。具体说明如下。

- static：静态的，无特殊定位。对象遵循 HTML 文档的定位规则。该项是默认值。
- absolute：绝对定位。将对象从文档流中拖出，将该对象相对于其最接近的一个具有定位设置的父对象进行绝对定位。如果不存在这样的父对象，则依据 body 对象进行定位。
- fixed：固定定位。它可以使元素对象脱离文档流固定在浏览器的某个位置。
- relative：相对定位。它可以使用 CSS 样式中的位置属性使元素对象在正常的文档流中偏移位置。



fixed 值在一些早期版本的浏览器(如 IE6)中并不提供对它的支持。

2. 元素位置

元素的位置属性可以配合元素的定位方式实现对页面元素位置的指定。该属性由 4 个属性构成：top、right、bottom、left，分别用于设置元素与顶部、右侧、下部、左侧的距离。

比如要将某个 id 为 ad 的元素固定到页面右下角，可以使用如下代码：

```
#ad {
    position:fixed;
    right:0;
    bottom:0;
}
```

11.2.2 实例描述

窗内网应广大网友要求，建立了几个技术讨论群。在这些 QQ 群里，大家可以实时地提出一些问题，供大家进行讨论。

初期为了让大家尽快加入这些群，能够进行技术讨论，需要将这几个群的号码放在窗内网首页进行公布。当然，最好的方法就是使用浏览器中固定不动的广告框来提示站点的浏览者。

下面我们就在窗内网首页的右下角添加一个显示 QQ 群号码的提示框。

11.2.3 实例应用

【例 11-1】具有固定位置的广告框

(1) 在 Dreamweaver 中创建一个站点，指定本地硬盘上的一个目录。

(2) 在站点中添加一个 imgs 文件夹和一个 css 文件夹。

(3) 在 css 目录中创建一个 CSS 样式表文件，命名为 position.css。

(4) 修改 position.css 文件，添加样式表代码，如下所示：

```
@charset "utf-8";
/* CSS Document */
```

```
#ad {
    background-image:url(../imgs/ad_bg.gif);
    height:126px;
    width:211px;
    bottom:0;
    right:0;
    position:fixed;
    padding:9px;
    font-family:"宋体", Arial;
}
#ad span {
    margin:0 0 0 15px;
    font-size:14px;
    font-weight:700;
    color:#FFF;
}
#ad ul li {
    color:#FFF;
    font-size:12px;
    padding:5px;
}
```

(5) 在站点根目录中添加一个名为 position.html 的 Web 页面文件。

(6) 打开 position 文件，在 head 部分引入刚才创建的 CSS 样式表文件，代码如下：

```
<link rel="stylesheet" type="text/css" href="css/position.css"/>
```

(7) 在页面 body 部分添加显示 QQ 群号信息的页面元素，代码如下：

```
<div id="ad">
  <span>窗内 QQ 群号</span>
  <ul>
    <li>QQ 群 1: 33925615</li>
    <li>QQ 群 2: 45368980</li>
    <li>QQ 群 3: 107423140</li>
    <li>QQ 群 4: 7858178</li>
  </ul>
</div>
```

11.2.4　运行结果

在 IE 浏览器中运行 position.html 页面，结果如图 11-2 所示。



图 11-2　固定位置广告框运行结果

11.2.5　实例分析



源码解析：

本实例定义了一个描述广告框样式的样式表。

首先设置广告框 ad 的背景、宽度、高度、字体、内填充，以及广告框的定位方式为固定定位，而且广告框右边和下边的位置都为 0，即紧贴浏览器内容框的边缘，这样该广告框将始终固定显示在浏览器的右下角。

然后样式表定义了广告框里的文本样式和边距等信息。

11.3　文字阴影效果

文字阴影效果是文本效果中最常用的效果之一。

使用阴影效果文本的通常做法是使用图片做成阴影效果引入页面。这样做非常不方便，特别是在文本内容替换的时候我们还需要手动处理图片文件。

其实还可以使用 CSS 样式表定义一个简单的文字阴影效果，下面我们来学习这种方法。



视频教学：光盘/视频/11/11.3　文字阴影效果.avi　　**长度：**11 分钟

11.3.1　基础知识——z-index 属性

CSS 样式定义的阴影效果其实是重叠着的两行文字，位置稍微相差一点即可实现阴影效果。

重叠的文字需要将相同大小的文字副本放置在不同的容器内部，并同时将容器的定位方式设置为相对定位。这样就可以实现不同的元素重叠放置。

但是，哪个容器在前面，哪个在后面，这个如果控制得不好，很可能显示出来的效果就大不一样。这里就要用到对层叠容器层叠顺序的设置，就是 CSS 中的 z-index 属性。

z-index 属性可以使用整数控制层叠元素显示的顺序，其格式如下：

```
z-index : auto | number
```

该属性默认值是 auto，不过我们可以设置一个整数值来控制显示层次。

在实际使用时要注意该参数的值可以设置为负数。如果是负数，容器中的内容将显示在页面内容的下面，就像水印效果。

比如页面中有一个 id 属性值为 div1 的 div 元素，我们在样式表中为其添加如下的 CSS 样式代码：

```
#div1 {
    position:absolute;
    top:60px;
    left:60px;
    background-color:#888;
    width:100px;
    height:100px;
    z-index:1;
}
```

理所当然，这个 div 将使用绝对定位的方式显示在页面中，其离上边和左边的距离都是 60 像素，该 div 的背景色为灰色，宽度和高度都是 100 像素，还有最重要的一点：它的 z-index 属性为 1，说明该元素将显示在其他元素上面，如图 11-3 所示。

当然，如果将上面代码中的 z-index 属性改为负数，比如-1，那么这个 div 将像一个水印一样显示在页面内容的下面，如图 11-4 所示。

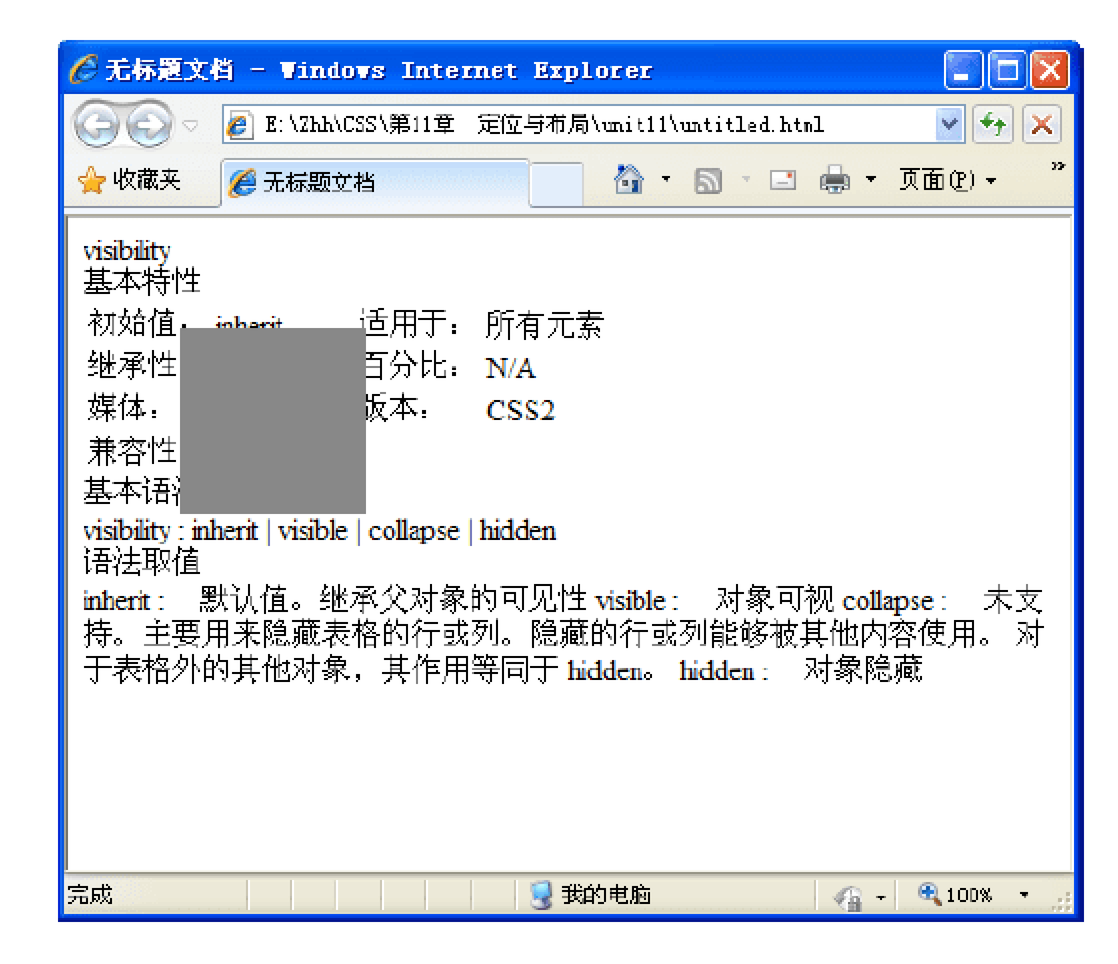


图 11-3　z-index 属性为 1

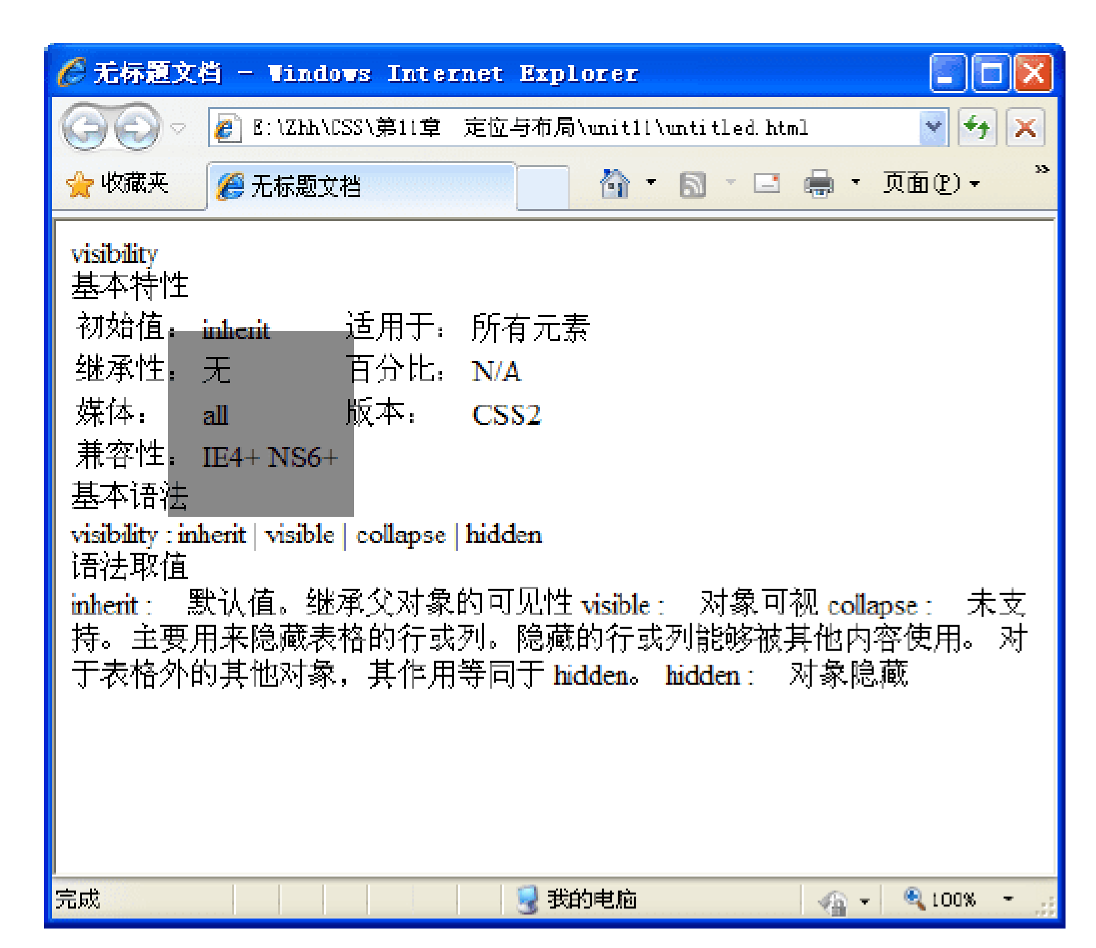


图 11-4　z-index 属性为-1

11.3.2 实例描述

以前好像在哪个国外的网站上见过使用 CSS 控制的页面重叠效果来实现它们站点的标题，整体风格看起来非常简洁、漂亮。刚才上网找了好久也没有找到这个网站，或许是它已经改版了吧。也许是使用这种方式比较生硬吧，但对实现页面局部的效果确实是一个不错的方法。

下面我们就来简单演示一下它的用法。

11.3.3 实例应用

【例 11-2】文字阴影效果

(1) 使用 Dreamweaver 打开上一节创建的站点。

(2) 新建一个 HTML 页面，命名为 zindex.html。

(3) 打开 zindex.html 页面并在页面中添加如下代码：

```
<div id="shadow">
  <div class="s1">ABC 阴影文本</div>
  <div class="s2">ABC 阴影文本</div>
</div>
```

(4) 在 css 目录中添加一个 CSS 样式表文件，命名为 zindex.css。

(5) 编辑 zindex.css 文件，添加样式表定义代码，如下所示：

```
@charset "utf-8";
body {
    background-image:url(../imgs/zindex_bg.jpg)
}
#shadow {
    position:relative;
    margin:100px 60px;
    font-size:60px;
    font-weight:900;
    font-family:"宋体", Arial;
}
#shadow .s1, #shadow .s2 {
    position:absolute;
}
#shadow .s1 {
    z-index:2;
    top:0;
    left:0;
    color:#999;
}
#shadow .s2 {
    z-index:1;
    top:3px;
    left:3px;
```

```
    color:#555;
}
```



这里必须定义两个重叠元素的父元素的定位方式，否则这两个元素将会脱离文档流造成页面混乱。

(6) 在 zindex.html 文件的 head 部分使用 link 关键字引入该样式表文件，其代码如下：

```
<link rel="stylesheet" type="text/css" href="css/zindex.css"/>
```

11.3.4 运行结果

使用 IE8 浏览器打开 zindex.html 页面，运行结果如图 11-5 所示。



图 11-5 阴影文本运行效果

11.3.5 实例分析



源码解析：

本实例在页面中放置了两个文本内容一样的 div 元素，class 属性分别为 s1 和 s2，用于分别设置颜色，并将其设置成重叠阴影效果。

接下来在样式表中定义文本样式，并设置两个 div 显示方式为绝对定位，实现重叠效果，并分别定位两个 div 元素的层叠顺序、位置和文本颜色，实现元素的错位显示，实现阴影效果。

11.4 文字环绕的图片

在网页布局中总会遇到一些横向布局的案例，两个或多个模块排成一排显示在页面中。这种效果如果使用表格实现则非常简单，我们可以把横排的模块放到一个表格的一行中即可。

这里我们不使用表格，而改为 div+CSS 的方式实现页面中的布局。我们知道 div 在页面中

显示的是一个占用整行的框，默认情况下每一个 div 都占用一行。我们要使多个 div 呈现在一行中，需要使用 CSS 控制它的显示方式，就是这里要讲的 float 属性。


视频教学：光盘/视频/11/11.4　文字环绕的图片.avi　　长度：11 分钟

11.4.1　基础知识——浮动定位

要实现元素脱离文档流浮动定位需要设置该元素的 float 属性。但是设置完 float 属性以后将会影响到页面的布局效果，所以在必要的时候还需要使用 clear 属性来清除元素的浮动效果。

1. float 属性

要实现元素的浮动定位需要设置该元素 CSS 样式的 float 属性。float 属性设置了页面对象在页面中是否浮动及如何浮动。该属性格式如下：

```
float : none | left | right
```

对于该属性取值的说明如下。

- none：对象不浮动。该项是默认值。
- left：对象靠容器左侧浮动。
- right：对象靠容器右侧浮动。

比如页面中有这样一段代码：

```
<div>
  <div id="div1">这是 div1 里的内容</div>
  <div id="div2">这是 div2 里的内容</div>
</div>
```

如果要使 div1 和 div2 在一行中显示，就可以设置 div1 或者它们两个的浮动属性，如下所示：

```
#div1, #div2 {
    float:left;
}
```



很多情况下浮动到一行的多个容器是同一类的项目，也经常是循环遍历出来的，所以我们通常直接设置它们所有项的浮动属性。

2. clear 属性

因为 float 属性是将元素移出文档流，所以在所有内容都浮动起来的时候页面内部的容器会认为自己已经是空容器，将仍然影响页面的效果。

比如我们修改上面的页面代码，如下所示：

```
<div id="div">
  <div id="div1">这是 div1 里的内容</div>
```

```
  <div id="div2">这是 div2 里的内容</div>
</div>
<span>这是 span 标签里的内容</span>
```

然后为其添加 CSS 样式，代码如下所示：

```
#div {
    border:solid 1px #ccc;
}
#div1, #div2 {
    float:left;
}
span {
    font-size:30px;
}
```

这时候我们期望的往往是 div1 和 div2 能够成一行显示，同时 div 的灰色边框将它们括起来，然后 span 元素中的文本在 div 下面独占一行地显示。

真是这样的么？答案：不是。

我们来看运行效果，如图 11-6 所示。

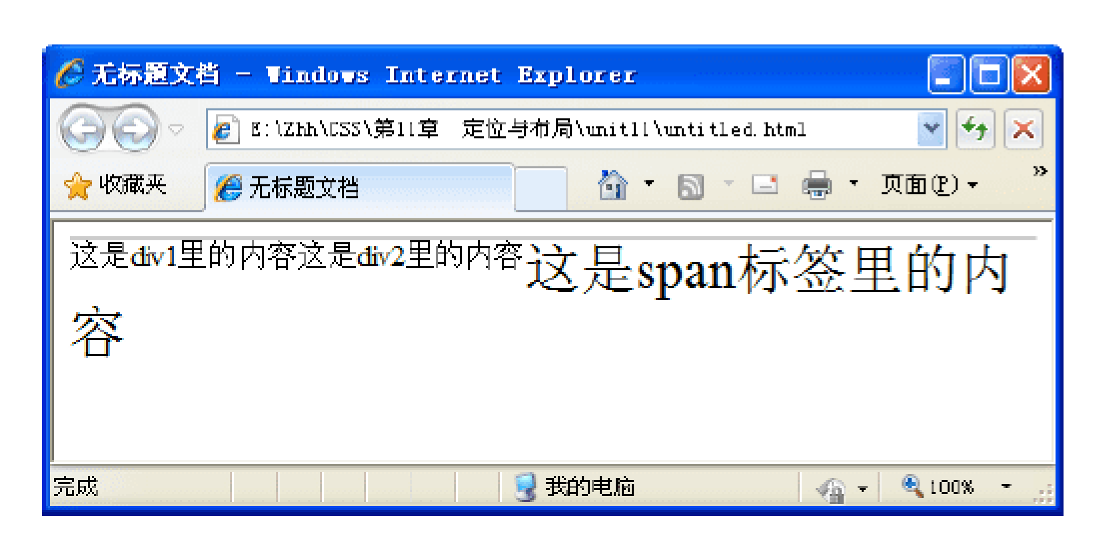


图 11-6　元素浮动的运行结果

要解决这个问题，我们就要使用另一个 CSS 样式属性——clear 属性。

clear 属性指定了当前元素不允许有浮动元素的边。如果该边有浮动元素，当前对象自动在该边外侧执行换行操作。

clear 属性的语法如下：

```
clear : none | left | right | both
```

对于该属性取值的说明如下。

- none：允许两边都有浮动对象。该项是默认值。
- left：不允许左边有浮动对象。
- right：不允许右边有浮动对象。
- both：两边都不允许有浮动对象。

要完善上面的实例，我们修改页面代码，如下所示：

```
<div id="div">
  <div id="div1">这是 div1 里的内容</div>
  <div id="div2">这是 div2 里的内容</div>
  <div style="clear:both;"></div>
```

```
</div>
<span>这是 span 标签里的内容</span>
```

我们在 div2 元素的后面又添加了一个元素，该元素设置 clear 样式为 both，说明该元素两边(特别是左边)都不允许有浮动对象，所以它自己会在 div2 后面单独占一行。这时父容器要正常显示最后那个没有浮动的 div，所以页面就会如我们预料的那样正常显示了，如图 11-7 所示。

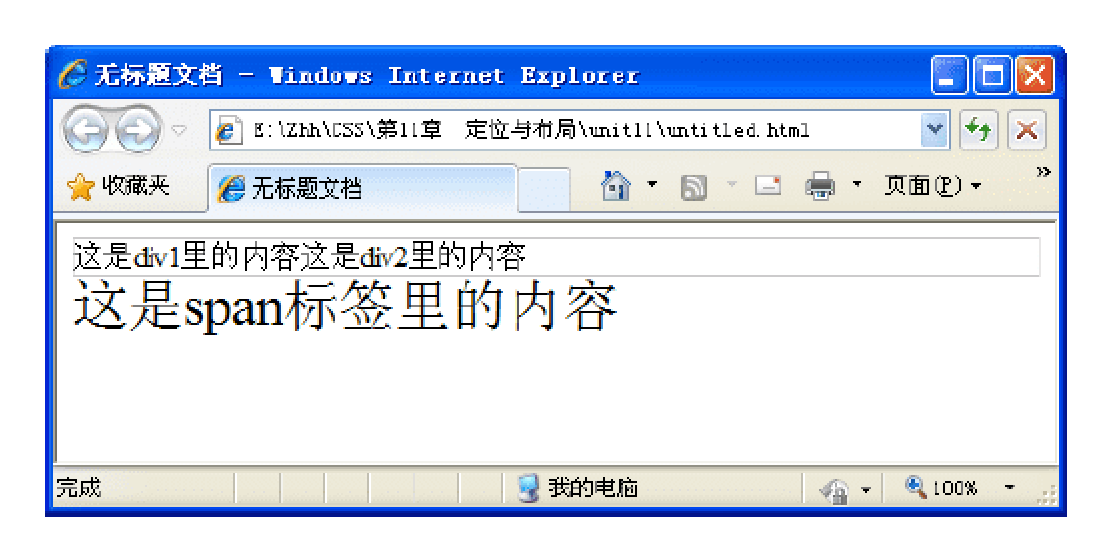


图 11-7　清除浮动以后的页面效果

这种方法虽然达到了效果，但是它需要修改页面结构，而这一点是 CSS 编码人员非常不愿做的一件事。本来追求的就是页面代码和样式分离，而现在为了添加样式而修改页面结构，怎么想都不合适。其实使用 CSS 样式表完全可以解决这个问题。

它不是需要在容器最后面添加一个有 clear 属性的元素么？我们可以使用 CSS 的 content 属性配合:after 伪类实现这个功能。

首先把添加的这一行代码去掉：

```
<div style="clear:both;"></div>
```

然后在样式表中为 id 属性为 div 的元素添加一个:after 伪类的样式，代码如下：

```
#div:after {
    content: ".";
    clear: both;
    visibility: hidden;
}
```

这段代码在容器内部最后面添加一个纯文本的元素，元素只有一个点。然后为该元素添加一个 clear 样式，值为 both。当然，该元素是为控制样式使用的，所以它不能影响页面文档的内容，最后我们还要把它隐藏，这里使用了 visibility 属性(这个属性稍后我们会讲到)。运行之后，一样可以让页面如我们预期的那样显示。

11.4.2　实例描述

像很多报纸上的版块一样，一篇新闻往往附带一张或多张图片，这些图片多是对新闻内容的一个说明。为了将适当的图片放置到适当的位置，而且既美观又节省页面资源，往往会使用文字环绕的效果布局页面。当然，很多网站或博客也大量使用这种方式布局网页。

环绕效果节省了页面中大量的空白，使页面显得更加紧凑。

下面我们就使用 CSS 控制页面内容实现一个文字环绕的图片效果。

11.4.3　实例应用

【例 11-3】文字环绕的图片

(1) 使用 Dreamweaver 打开上一节我们使用过的站点。

(2) 新建一个 HTML 文件，命名为 float.html。

(3) 修改 float.html 文件，添加如下页面代码：

```
<div id="content">
  <img src="imgs/box.jpg" class="box" />
  <p>    在 Web 页面中的“盒子”也是同样的结构：厚度、边距(边缘与其他物体的距离)、填充(填充厚度)。引申到 CSS 中，就是 border、margin 和 padding。当然，不能少了内容。也就是说整个盒子在页面中占的位置大小应该是内容的大小加上填充的厚度再加上边框的厚度再加上它的边距。</p>
  <p>    在这里我们需要牢记于心的是，现实中我们使用盒子是为了包装和保护物品，而在 Web 页面中使用盒子，完全是为了美观。所以理解“盒子模型”的时候不要对它有什么“敬畏”之心。
    因为 HTML 页面是一个层级的“树”，所以盒子的嵌套使用是很正常的，就像平时我们使用盒子分门别类地存放东西，最后放到一个大箱子里一样。</p>
  <p>    总体来说，使用“盒子模型”这个名称还是非常形象的。它和现实中的盒子非常相似。如果你能把现实世界中的三维的盒子抽象到二维平面中，你就已经理解什么是“盒子模型”了。</p>
</div>
```

(4) 在 css 目录中创建一个 CSS 文件，命名为 box.css。

(5) 编辑 box.css 文件，添加如下的 CSS 样式代码：

```
@charset "utf-8";

body {
    background-image:url(../imgs/float_bg.gif);
    padding:70px 0 0 50px;
}
#content {
    font-size:13px;
    font-family:"宋体", Arial;
    color:#666;
    width:550px;
    line-height:21px;
}
#content .box {
    float:right;
    width:206px;
    height:146px;
    margin:10px;
}
```

(6) 在 float.html 页面的 head 部分引入 box.css 文件，代码如下：

```
<link rel="stylesheet" type="text/css" href="css/box.css"/>
```

11.4.4 运行结果

使用 IE8 浏览器打开 float.html 页面，运行结果如图 11-8 所示。



图 11-8 文字环绕效果

11.4.5 实例分析



源码解析：

本实例首先在容器 div 中加入一幅图片和三段文本，然后在 CSS 文件中设置页面的背景；以及容器元素的字体、字号、颜色、宽度、行高等样式；以及设置容器中图片的宽度、高度、边距和向右浮动的属性。最终达到图片靠右侧浮动，文字在图片左侧环绕显示的效果。

11.5 显示一个图片的局部

有时页面中实现一些特殊效果需要使一些元素只显示一部分内容，比如放大镜效果。

通常放大镜效果需要在缩略图中指定显示的区域，然后在页面上浮动一个层显示大图的部分内容。这就要用到 CSS 样式表中的一个属性——clip。



视频教学：光盘/视频/11/11.5 显示一个图片的局部.avi　　　　长度：11 分钟

11.5.1 基础知识——clip

显示一个元素的局部内容需要用到一个非常实用的 CSS 样式属性：clip。

clip 属性可以设置页面中元素的可视区域，区域外的部分是透明的。该属性的语法如下：

```
clip : auto | rect ( number number number number )
```

对于该属性两个可选值的说明如下。

- auto　对象无剪切。默认值，即显示元素的所有区域
- rect (number number number number)　根据上、右、下、左的顺序提供对象自左上角为(0, 0)点的坐标计算的 4 个数值。4 项中的任意一项都可以使用 auto 替换，即表示该边不进行裁切。

使用该属性需要注意一点，只有被设置元素的 position 属性的值为 absolute 时，clip 属性设置的值才起作用。

11.5.2　实例描述

实现一个放大镜效果稍微有点复杂，需要使用 JavaScript 控制元素显示的区域。这里我们只是学习 clip 属性的用法，就没有必要做得那么复杂。所以这里我们简单地使用 clip 属性“切”出一个图片的局部内容显示在页面中。

11.5.3　实例应用

【例 11-4】显示一个图片的局部

(1) 使用 Dreamweaver 打开 11.4 节中使用的站点。

(2) 在 imgs 文件夹中准备一个图片文件 flower.jpg，以备在页面中使用 clip 属性裁切它。完整的图片如图 11-9 所示。



图 11-9　图片 flower.jpg

(3) 在站点根目录下创建一个名为 clip.html 的 HTML 文件。

(4) 在 clip.html 页面中将前面准备的图片文件引入，代码如下：

```
<img src="imgs/flower.jpg" id="flower" />
<img src="imgs/flower.jpg" style="margin:20px 0 0 320px; width:260px;"  />
```

(5) 在 css 目录中添加一个 CSS 资源文件，命名为 clip.css，为页面添加 CSS 样式，代码如下：

```
@charset "utf-8";

body {
    background-image:url(../imgs/clip_bg.jpg)
}
#flower {
    position:absolute;
    top:50px;
    left:30px;
    clip:rect(auto 230px 160px 90px)
}
```

(6) 在 clip.html 页面的 head 部分引入 CSS 文件，代码如下：

```
<link rel="stylesheet" type="text/css" href="css/clip.css"/>
```

11.5.4 运行结果

使用 IE8 浏览器打开 clip.html 页面，运行结果如图 11-10 所示。

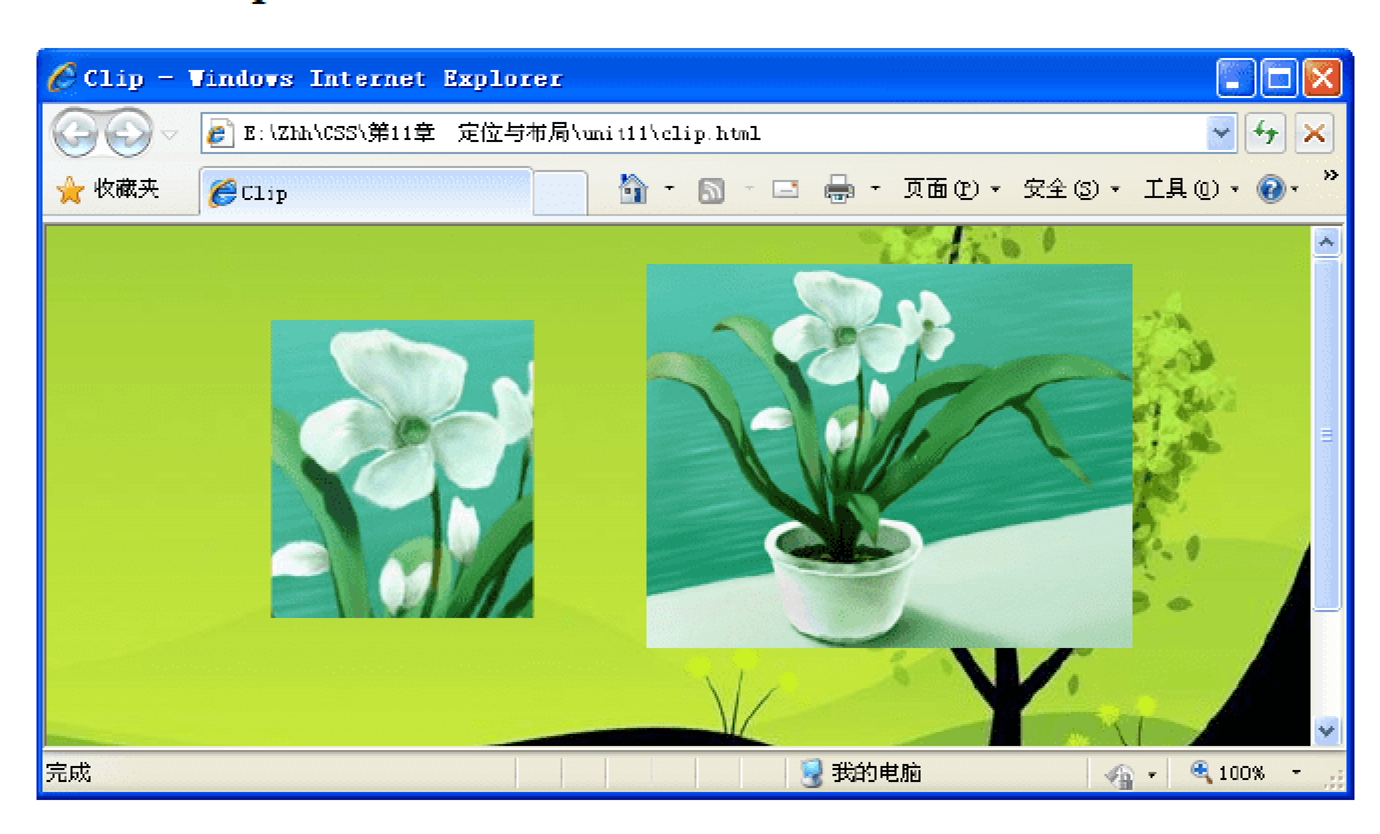


图 11-10 clip 属性效果

11.5.5 实例分析



源码解析：

本实例先在站点中准备一张图片，然后在页面中引入该图片，添加 ID 属性为 flower。接下来为页面中元素添加 CSS 样式代码，其中设置页面中 flower 元素的定位样式为 absolute(绝对定位)，并设置其顶部位置为 50 像素，左侧位置为 30 像素，最后使用 clip 属性设置元素的可见区域为顶部自动(无裁切)、右侧到 230 像素位置、底部到 160 像素位置、左侧到 90 像素的位置，该范围以内的为元素可见区域。

11.6　带滚动条的页面模块

页面模块中的内容往往大小不固定，比如有时内容多，有时内容少，所以往往页面中很难控制模块的样式。不过对于这一点，CSS 也给出了相当有伸缩性的处理方式。

CSS 专门提供了一个处理容器内容太多而溢出的样式属性——overflow。

**视频教学：**光盘/视频/11/11.6　带滚动条的页面模块.avi　　长度：11 分钟

11.6.1　基础知识——overflow

overflow 属性用于设置容器在内容太多而造成溢出的情况下的行为。该属性的语法如下：

```
overflow : visible | auto | hidden | scroll
```

关于该属性的值说明如下。

- visible：显示所有内容。设置该属性时容器将随内容增大而变大，这样可能会影响预期的页面效果。
- auto：自动。当内容超出显示区域时为容器添加滚动条。
- hidden：不显示超出容器显示区域的内容。
- scroll：总是显示滚动条。

另外，CSS 还提供了两个扩展的属性：overflow-x 和 overflow-y。这两个属性分别用于对 X 方向(横向)和 Y 方向(纵向)的溢出处理方式进行设置。

11.6.2　实例描述

有时候，我们在设计一个 Web 页面时会过多地考虑页面美观的问题，而把页面中的某个模块设计得比较小。但是也有的时候这个小模块中需要放置较多的内容，这样就会造成一些矛盾：如果固定模块的大小，那么内容将不能完全显示；而如果显示出所有的内容，那么将会影响页面美观。

解决这个问题的方法就是使用滚动条：让页面模块大小固定，让用户使用滚动条来翻动查看模块里的内容。

下面我们就来使用滚动条在一个比较小的模块中显示一大段文本内容。

11.6.3　实例应用

【例 11-5】带滚动条的页面模块

(1) 使用 Dreamweaver 打开 11.5 节使用的站点。

(2) 在站点根目录中新建一个 HTML 页面，命名为 overflow.html。

(3) 在页面中添加页面内容，代码如下：

```
<div id="content">
    很多时候我们设计得非常漂亮的页面中某个模块因为美观等原因设计的比较小，或许是可能觉得放置的内容不会太多，也可能是页面设计者太注重美观效果了。所以我们不得不在有限的大小中显示足够多的内容，这个问题的一个比较不错的解决方案就是设置滚动条。让模块框大小固定，但是内容可以滚动显示，两全其美。
</div>
```

(4) 在 css 目录中创建一个 CSS 样式表文件，命名为 overflow.css。

(5) 编辑 overflow.css，添加样式代码，如下所示：

```
@charset "utf-8";

body {
    background-image:url(../imgs/overflow_bg.jpg)
}
#content {
    border:solid 1px #693;
    width:200px;
    height:120px;
    padding:8px;
    margin:160px 0 0 260px;
    font-size:13px;
    font-family:"宋体", Arial;
    line-height:18px;
    color:#060;
    overflow:auto;
}
```

(6) 在 overflow.html 页面的 head 部分引入 css 样式表文件，代码如下：

```
<link rel="stylesheet" type="text/css" href="css/overflow.css"/>
```

11.6.4 运行结果

使用 IE8 浏览器打开 overflow.html 页面，运行结果如图 11-11 所示。



图 11-11 overflow 运行结果

11.6.5　实例分析



源码解析：

本实例首先创建一个 HTML 页面文件，并添加页面元素，然后再创建一个样式表文件，为页面中的元素添加 CSS 样式。

在样式表文件中，首先设置页面的背景，然后设置页面模块 content 的边框、大小、填充、边距、字号、字体、行高、颜色等属性，然后再使用 overflow 属性将 content 模块的溢出处理方式设置为自动，这样在模块内容溢出时将会自动为模块添加滚动条。

11.7　横排的列表元素

正常情况下，HTML 中列表的显示是纵向排列的，不过有些时候往往需要将它横向排列，比如设计一个页面的菜单。

虽然横向排列的页面菜单可以有多种实现方式，但是很多人仍喜欢使用 ul 和 li 元素来定义这个列表。

其实实现使用 ul 和 li 元素定义的列表横向显示也不只有一种方式，不过这里我们只讲其中最简单的一种——使用 display 属性实现。



视频教学：光盘/视频/11/11.7　横排的列表元素.avi　　　　长度：4 分钟

11.7.1　基础知识——display

display 属性可以设置页面元素的显示方式。该属性的语法如下：

```
display : block | none | inline | compact | marker | inline-table | list-item
| run-in | table |table-caption | table-cell | table-column | table-column-group
| table-footer-group | table-header-group | table-row | table-row-group
```

关于该属性的可选值说明如下。

- block：块对象。将该元素强制作为块对象呈现到页面中。
- none：隐藏对象。设置该项的元素将被隐藏。
- inline：内联对象。使用该选项将从对象中删除行，即多个元素在同一行中显示。
- compact：分配对象为块对象或基于内容之上的内联对象。
- marker：指定内容在容器对象之前或之后。要使用此参数，对象必须和:after 及:before 伪元素一起使用。
- inline-table：将表格显示为无前后换行的内联对象或内联容器。
- list-item：将块对象指定为列表项目。并可以添加可选项目标志。

- run-in：分配对象为块对象或基于内容之上的内联对象。
- table：将对象作为块元素级的表格显示。
- table-caption：将对象作为表格标题显示。
- table-cell：将对象作为表格单元格显示。
- table-column：将对象作为表格列显示。
- table-column-group：将对象作为表格列组显示。
- table-footer-group：将对象作为表格脚注组显示。
- table-header-group：将对象作为表格标题组显示。
- table-row：将对象作为表格行显示。
- table-row-group：将对象作为表格行组显示。

在上面所有属性值选项中，我们经常会使用 block 将一个 span 元素呈现为块对象，或者使用 none 将元素直接隐藏……

不过本节我们要将一个列表中的所有项显示到一行中，就可以使用 inline 项来实现这个效果。代码如下：

```
ul li {
    display:inline;
}
```

11.7.2 实例描述

横向的列表元素效果经常用于设计页面菜单条。确实，使用这种方式设计出来的菜单条结构比较清晰，更容易阅读和维护。

本实例我们就使用 CSS 配合 ul 及 li 元素设计一个漂亮的菜单条。

11.7.3 实例应用

【例 11-6】横排的列表元素

(1) 使用 Dreamweaver 打开 11.6 节使用的站点。

(2) 新建一个 HTML 页面，命名为 display.html。

(3) 编辑 display.html 页面，添加页面内容，代码如下：

```
<div id="menubar">
  <ul>
    <li><a href="#"><span>Home</span></a></li>
    <li><a href="#"><span>MyPhoto</span></a></li>
    <li><a href="#"><span>MyArticle</span></a></li>
    <li><a href="#"><span>MyFamily</span></a></li>
    <li><a href="#"><span>MyScroll</span></a></li>
    <li><a href="#"><span>MyFriend</span></a></li>
     <li><a href="#"><span>Message</span></a></li>
```

```
  </ul>
</div>
```

(4) 在 css 目录中创建一个 CSS 样式表文件，命名为 display.css。

(5) 编辑 display.css 文件，添加样式代码，如下所示：

```
@charset "utf-8";

body {
    background-image:url(../imgs/display_bg.jpg)
}
ul, li, a {
    margin:0;
    padding:0;
}
#menubar {
    margin:180px 0 0 0;
    height:24px;
    font-size:14px;
    border-bottom:1px solid #E276A7;
}
#menubar li {
    display:inline;
}
#menubar a {
    float:left;
    background:url("../imgs/m left.gif") no-repeat left top;
    padding:0 0 0 4px;
    text-decoration:none;
}
#menubar a span {
    display:block;
    background:url("../imgs/m_right.gif") no-repeat right top;
    padding:5px 15px 4px 6px;
    color:#333;
}
```

(6) 在 display.html 页面的 head 部分添加对 CSS 样式表文件 display.css 的引用，代码如下：

```
<link rel="stylesheet" type="text/css" href="css/display.css"/>
```

11.7.4　运行结果

使用 IE8 浏览器打开 display.html 页面，运行结果如图 11-12 所示。

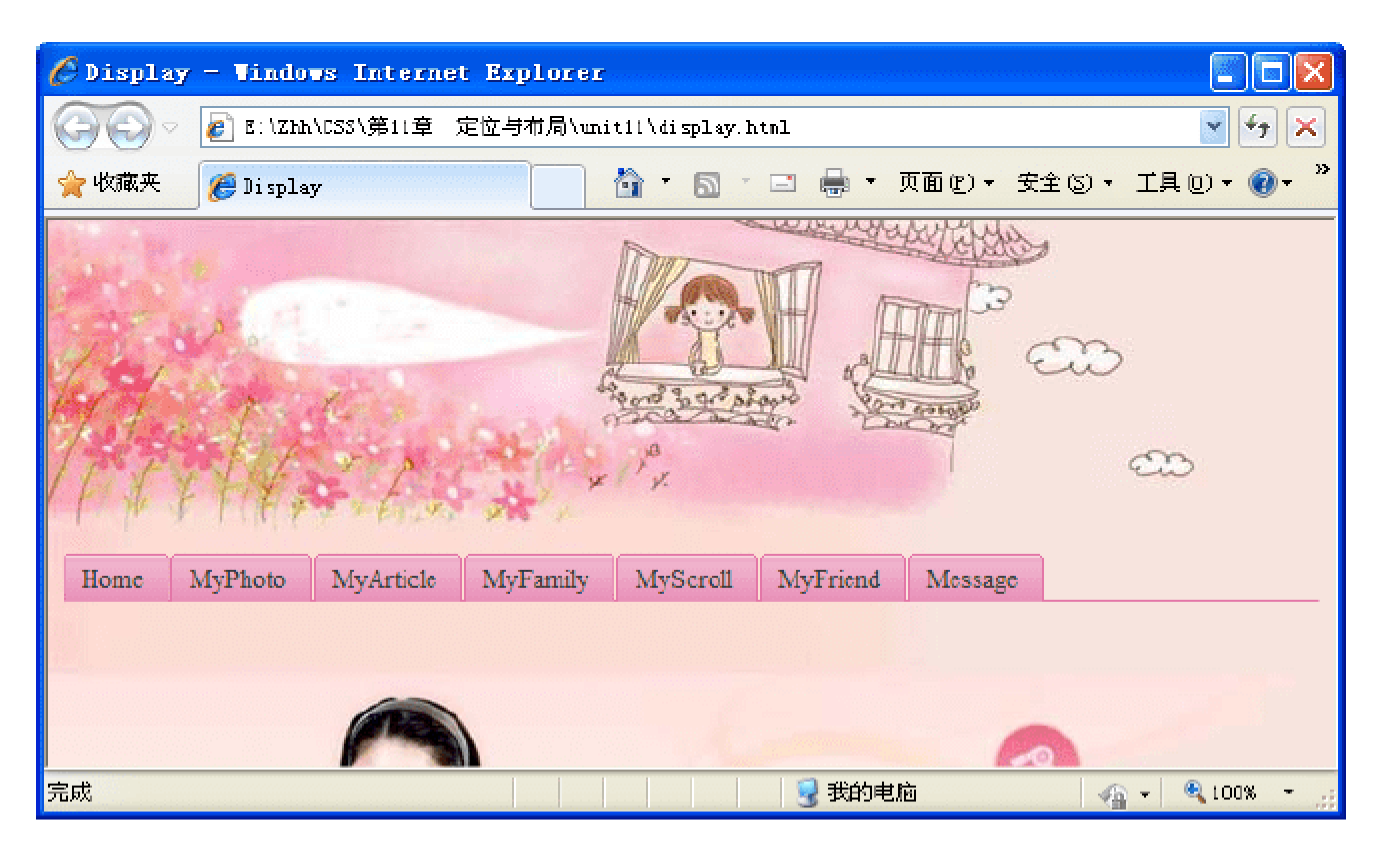


图 11-12　横排的列表元素

11.7.5　实例分析



源码解析：

本实例首先在页面中添加一个作为菜单条容器的 div 元素，用于控制页面中菜单条的位置；然后在该 div 中添加一个用于排列菜单项的列表元素，以及菜单项里的内容；最后在样式表文件中设置菜单条、菜单项，以及链接文本的样式。其中设置菜单项列表中的各项的 display 属性值为 inline 来使所有的菜单项显示在一行中，实现了一个横向排列的列表效果。

11.8　使页面元素不可见

要使页面元素不可见有几种方法，不过常用的是使用 CSS 样式表中的 visibility 属性或 display 属性设置页面元素的可见性。

上一节我们已经了解到了可以将 display 属性的值设为 none 来将页面中元素隐藏。不过还有一种方式，就是使用 visibility 属性控制页面元素的可见性。

visibility 属性的基本语法如下：

```
visibility : inherit | visible | collapse | hidden
```

对于该属性值的 4 个可选项，具体说明如下。

- inherit：默认值。继承父对象的可见性。
- visible：对象可见。
- collapse：主要用来隐藏表格的行或列。隐藏的行或列能够被其他内容使用。对于表格外的其他对象，其作用等同于 hidden。
- hidden：对象隐藏。

既然都是隐藏，两个不同的方法肯定应该有一定的差异。我们来看一下使用效果上的对比。比如页面中有三个 div，代码如下：

```
<div id="first">第一个 DIV</div>
<div id="second">第二个 DIV</div>
<div id="third">第三个 DIV</div>
```

在样式表中给它们添加样式：

```
div {
    border:solid 1px #ccc;
    margin:8px;
    padding:5px;
}
```

正常情况下浏览器中的运行效果如图 11-13 所示。

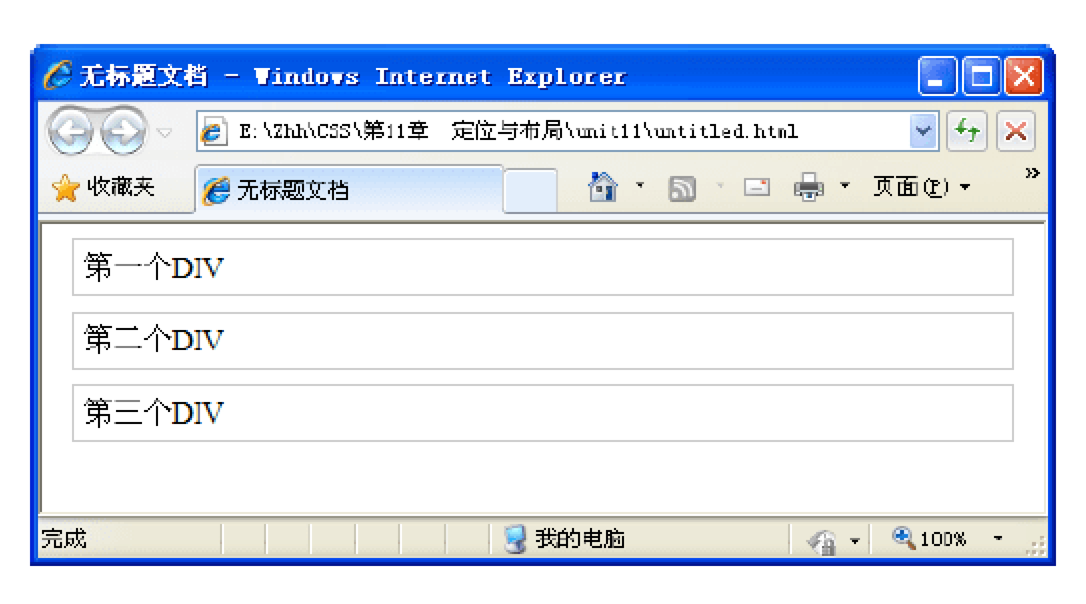


图 11-13　无隐藏效果

然后我们为第二个 div 添加 display 属性使其隐藏，代码如下：

```
#second {
    display:none;
}
```

运行以后，结果如图 11-14 所示。

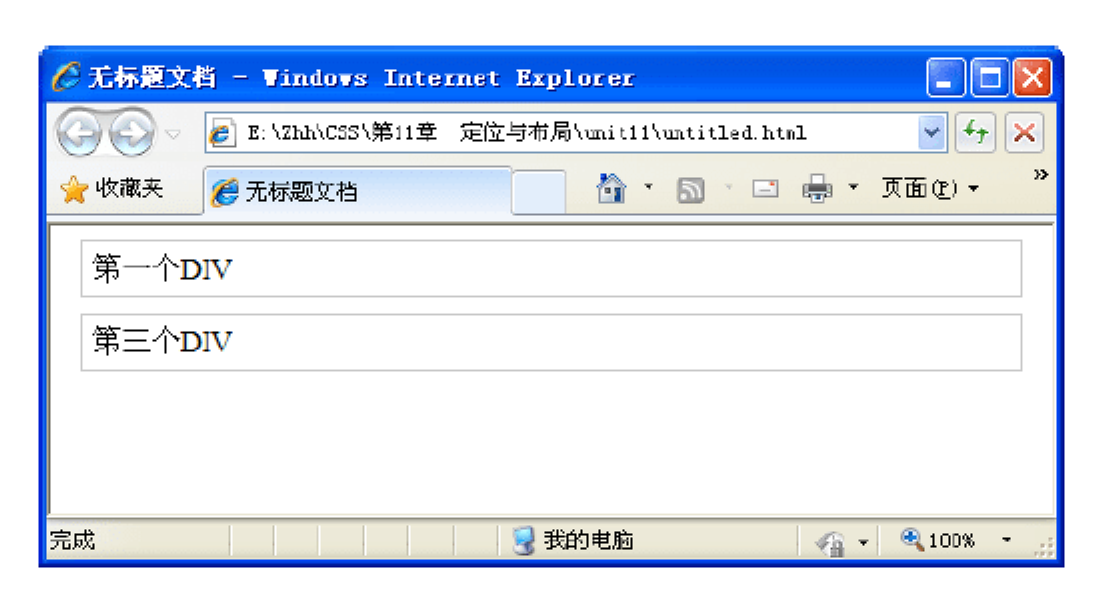


图 11-14　使用 display 属性隐藏元素

下面将 display 属性修改为使用 visibility 属性，代码如下：

```
#second {
    visibility:hidden;
}
```

运行以后，结果如图 11-15 所示。



图 11-15　使用 visibility 属性隐藏元素

通过上面的比较，就很容易看出使用 visibility 属性和 display 属性隐藏元素的差异了。visibility 属性只是设置该元素不可见，但是在文档中还保留着它的位置；而使用 display 属性则是将该元素从文档流中清理掉。具体用法视情况而定。

11.9　常见问题解答

11.9.1　div 布局用绝对定位好还是用相对定位好



div 布局用绝对定位好还是用相对定位好?
网络课堂：http://bbs.itzcn.com/thread-11906-1-1.html

请教大家 div 布局用绝对定位好还是用相对定位好?

【解决办法】使用 div+CSS 布局很少会用到定位，除非有需要做一些特殊效果的时候才需要定位。通常情况下，div 的布局就像表格那样嵌套使用。

如果非要问用绝对定位好还是用相对定位好？我个人觉得各有优劣，不过相对定位比较常用。

因为绝对定位和文档流没关系，如无声明，则其定位是相对 body 而言的。处理不好的话，如浏览器尺寸变了，页面可能就会变形。

相对定位的元素属文档流，所以是稳定的，相对定位是相对它应该出现的位置，如没有设置 top left 之类的属性，和普通的 div 一模一样。

绝对定位是这样用的：如果父元素是相对定位的，并且具有具体的布局，那么其子元素可以使用绝对定位（相对于父元素）。如果父元素不具有上述条件，那么子元素使用的绝对定位实际上是相对 body 进行定位的。

11.9.2　CSS 样式中 float 属性的问题



CSS 样式中 float 属性的问题。
网络课堂：http://bbs.itzcn.com/thread-11907-1-1.html

比如有如下代码：

```
<div id="content1">
  <div style="float : left"></div>
  <div style="float : right"></div>
  <div sytle="clear : both"></div>
</div>
```

一直搞不明白，既然清除了浮动，就不应该再浮动了，即两个 div 应该各自独占一行，而事实是两个 div 仍然浮动在左右两边？

【解决办法】可能是你理解错了。

清除浮动是指清除其他元素浮动对它造成的影响。帮助文档上说“该属性的值指出了不允许有浮动对象的边”，意思就是指定哪条边不可以有设置浮动属性的对象，如果有的话，就换一行显示，不跟它们一起显示。

所以说它设置的清除浮动不会对前两个元素的浮动属性造成影响。

11.10　习　　题

1. 填空题

(1) 如果要让一个 div 脱离文档流，显示在页面中的固定位置，可以设置该 div 元素 CSS 样式中的 position 属性值为__________。

(2) 如果要设置一个绝对定位的元素距页面底部的距离，可以设置该元素 CSS 样式表中的__________属性。

(3) 如果要设置一个绝对定位的图片的显示区域，可以设置该图片元素在 CSS 样式表中的__________属性。

(4) 如果要将一个超链接标签显示为块元素，需要将 CSS 样式表中的 display 属性的值设为__________。

2. 选择题

(1) 如果需要在容器元素内容溢出的时候为容器添加滚动条，可以设置该容器元素 CSS 样式 overflow 的值为__________。

A. visible　　B. auto　　C. hidden　　D. scroll

(2) 如果页面中一个 div 元素被设置了以下样式：

```
#div {
    width:100px;
    height:100px;
    position:absolute;
    top:0;
    left:0;
    right:0;
    bottom:0;
}
```

那么该元素将显示在__________。

A. 页面左上角　B. 页面右下角　C. 窗口右上角　D. 窗口左下角

(3) 如果要使页面中的一个元素不显示，而且还需要保留该元素所占的空间，可以设置该元素__________。

A. display 属性值为 none　B. display 属性值为 hidden

C. visibility 属性值为 none　D. visibility 属性值为 hidden

(4) 如果要设置多个 div 元素显示在一行中，可以设置这些元素 CSS 样式的__________属性为 left。

A. float　B. display　C. position　D. overflow

3. 上机练习

上机练习：实现页面固定位置工具条

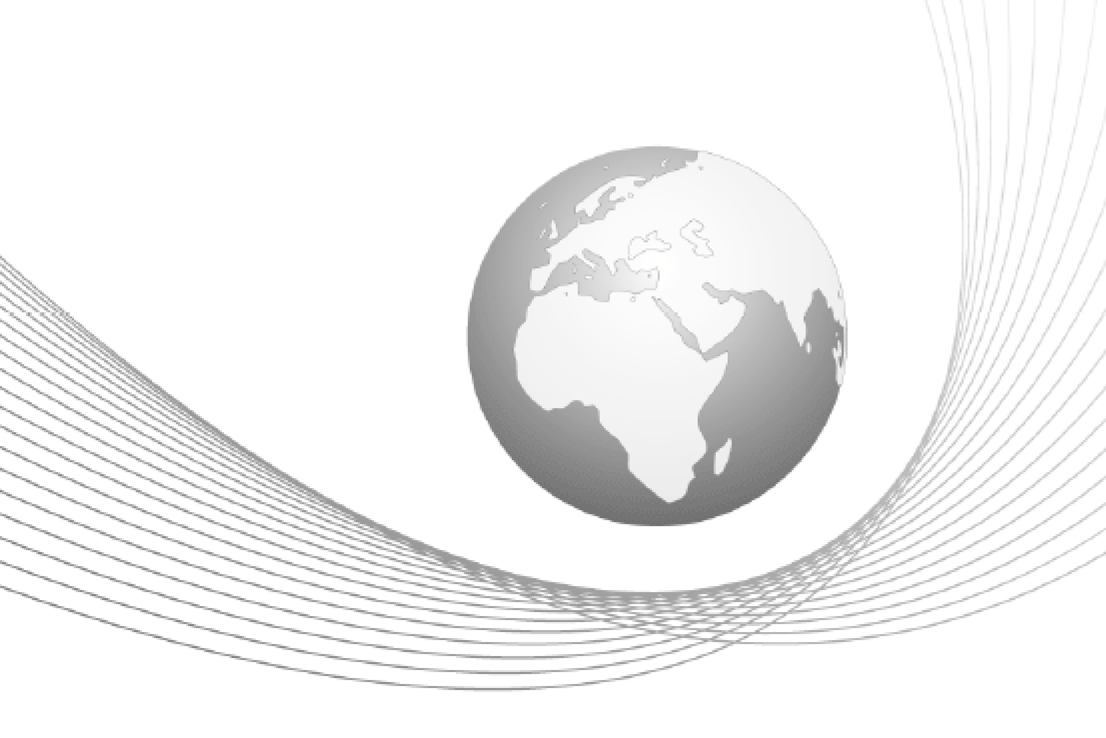
在有些空间或者论坛中会有一些显示在窗口内容区顶部或底部的一个工具条，显示一些常用的操作。

这个工具条在页面中滚动条滚动的时候一直保持在页面的顶部或底部，以方便我们对页面进行一些操作。

本练习我们实现一个固定在页面顶部的工具条，如图 11-16 所示。



图 11-16　固定位置的工具条



第 12 章　逐步改善页面的表格

内容摘要：

本章介绍一些改善页面表格显示布局的属性，如 border-collapse 属性设置表格的边框是否合并为单线边框、border-spacing 属性设置表格单元格之间的距离、caption-side 属性设置表格标题位置。通过本章的学习可以使页面表格更加简洁，更能够方便用户阅读表格数据。

学习目标：

- 了解表格特有的元素
- 掌握 border-collapse 属性的应用
- 掌握 border-spacing 属性的应用
- 掌握 caption-side 属性的应用
- 理解 empty-cells 属性的应用
- 理解 table-layout 属性的应用

12.1 表格特有的元素

如果数据表格对于视力正常的用户来说都难以阅读，那么可以想象对于必须使用辅助工具(如屏幕阅读器)的用户，就更加困难了。不过(X)HTML 规范提供了许多元素和属性来提高数据表格对于这些设备的可访问性。这些元素和属性当前并非都得到了屏幕阅读器的支持，但是尽可能使用它们肯定是好做法。

1．caption 和 summary

caption 元素是用做表格的标题。虽然它不是表格的必须元素，但是也应尽可能使用它为表格设置一个标题，便于用户阅读理解表格内容。

summary 属性应用于<table>标签，来描述表格的内容。它与 image 的 alt 文本相似，summary 应该总结表格中的数据，一个好的 summary 可以减少用户阅读表格数据的需要。

下面看看 caption 和 summary 如何应用在表格定义中，代码如下所示：

```
<table summary="学员信息表">
    <caption>学员信息表</caption>
</table>
```

2．thead、tbody 和 tfoot

使用 thead、tbody 和 tfoot 元素能够将表格划分为三部分，即头部、主体和尾部。可以将所有列标题放在 thead 元素中，这样就可以对这个特殊区域单独地应用样式。要使用 thead 元素或 tfoot 元素，那么就必须至少使用一个 tbody 元素。在一个表格中可以有多个 tbody 元素将复杂的表格划分为更容易管理的部分，但一个表格中只能有一个 thead 或 tfoot 元素。

3．col 和 colgroup

虽然使用 tr 元素能够对整行应用样式，但是很难对整列应用样式，因此 W3C 引入了 colgroup 和 col 元素。colgroup 能够对使用 col 元素定义的一个或多个列进行分组。但是，支持 col 和 colgroup 元素的样式的浏览器不多。

```
<colgroup>
    <col id="c1"/>
    <col id="c2"/>
    <col id="c3"/>
</colgroup>
```

12.2 制作一个单线边框表格

12.1 节介绍了表格的一些特有元素，使用这些元素可以在页面中定义一个简单的表格，本节将介绍用 border-collapse 属性设置表格的边框样式。

视频教学：光盘/视频/12/12.2　制作一个单线边框表格.avi　　长度：7 分钟

12.2.1　基础知识——border-collapse

border-collapse 属性可以设置表格的边框是否合并为一个单线边框，还是在标准的 HTML 中显示一个分开边框。此属性对于 currentStyle 对象而言是只读的。对于其他对象而言是可读写的。对应的脚本特性为 borderCollapse。该属性的可选属性值如表 12-1 所示。

表 12-1　border-collapse 属性值

属 性 值	说　明
separate	默认值。边框会被分开。不会忽略 border-spacing 和 empty-cells 属性
collapse	如果可能，边框会合并为一个单一的边框。会忽略 border-spacing 和 empty-cells 属性
inherit	规定应该从父元素继承 border-collapse 属性的值

border-collapse 属性的基本语法格式：

```
border-collapse : separate | collapse
```

定义两个表格，分别设置 border-collapse 属性为 collpase 和 separate，代码如下所示：

```
.tablesep
{
    border-collapse: separate       //设置边框为分开
}
.tablecoll
{
    border-collapse: collapse       //设置边框合并为一个单线边框
}
```

在 IE 浏览器中执行效果如图 12-1 所示。



图 12-1　设置表格边框是否合并

12.2.2 实例描述

我们在浏览网页时，经常看到这种效果：网页中的表格都是单线边框样式，这种样式给人的感觉是表格简洁大方，而在标准的 HTML 中显示的是一个分开边框。怎样才能使我们的表格边框成为单线边框呢？本节我们将制作一个单线边框表格的实例。

12.2.3 实例应用

【例 12-1】制作一个单线边框表格

(1) 新建一个 HTML 页面 border-collapse-exp.html，在该页面中定义一个表格，代码如下所示：

```
<form name="searchuser" method="POST">
    <center>
        <table width="80%" border="1" align="center" bordercolor="#C0C0C0"
class="table_form">
            <tr height=25  bgcolor="#E8E8E8">
                <th width="20%" align="center"><strong>编号</strong></th>
                <th width="20%" align="center"><strong>产品名称</strong></th>
                <th width="20%" align="center"><strong>所属分类</strong></th>
                <th width="20%" align="center">
                    <input  type='checkbox' name='chkall'><strong>全选
</strong>
                    <input type='submit' value='删除'>
                </th>
            </tr>
            <tr>
                <th>1</th>
                <th>女士羽绒服</th>
                <th>女士冬装</th>
            </tr>
            <tr>
                <th>2</th>
                <th>雪地靴</th>
                <th>冬靴</th>
            </tr>
        </table>
    </center>
</form>
```

(2) 使用 border-collapse 属性设置 table 元素的边框样式，代码如下所示：

```
<style type="text/css">
.table_form {
    border-collapse: collapse      //设置表格为单线边框
}
```

```
.table_form td, .table_form th {        //设置表格 td、th 样式
    line-height:150%;
    padding:4px;
    font-size:12px;
    font-family:"宋体";
}
.table_form th strong {
    color:#077ac7;
}
</style>
```

12.2.4　运行结果

在 IE 浏览器中打开 border-collapse-exp.html 文件，执行效果如图 12-2 所示。

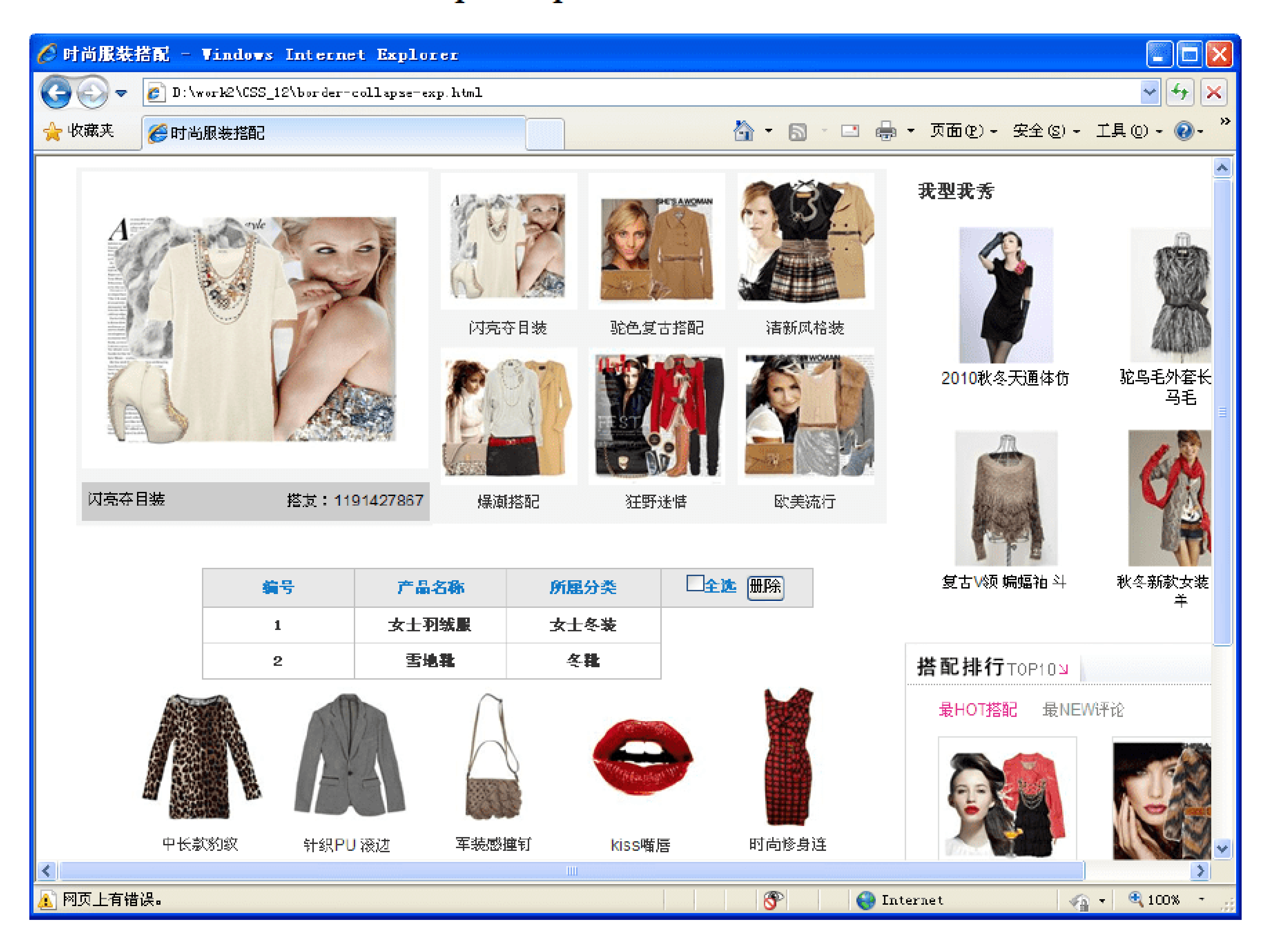


图 12-2　女士服饰表格

12.2.5　实例分析



源码解析：

本实例首先在 HTML 页面 border-collapse-exp.html 中定义了一个存储女士服饰数据的表格，然后为了使表格更加美观，使用 border-collapse 属性设置了表格的边框为合并的单线框，最后还设置了表格的 th、td 元素的字体、间隙等样式。

12.3 设置单元格边界之间的距离

若表格单元格边界之间的距离设置得当，可以使表格看起来更加简洁美观，同时也便于用户阅读表格数据。

**视频教学：**光盘/视频/12/12.3 设置单元格边界之间的距离.avi　　长度：4 分钟

12.3.1 基础知识——border-spacing

border-spacing 属性用于指定分隔边框模型中单元格边界之间的距离。当指定了两个长度值时，第一个是水平间隔，第二个是垂直间隔。除非 border-collapse 被设置为 separate，否则将忽略这个属性。border-spacing 属性的可选属性值如表 12-2 所示。

表 12-2 border-spacing 属性值

属性值	说明
length length	指定相邻单元格的边框之间的距离。可以使用 px、cm 等单位，但不允许使用负值。如果定义一个 length 参数，那么定义的是水平和垂直间距。如果定义两个 length 参数，那么第一个设置水平间距，而第二个设置垂直间距
inherit	规定应该从父元素继承 border-spacing 属性的值

border-spacing 属性的基本语法格式如下所示：

```
border-spacing : length
```



某些版本的 IE 浏览器不支持此属性。另外如果已规定!DOCTYPE，那么 Internet Explorer 8(以及更高版本)支持 border-spacing 属性。

12.3.2 实例描述

前段时间看到同事做了一个人事管理系统，她做的显示员工基本信息的表格看起来很漂亮，后来才知道，她只是设置了单元格边界之间的距离，现在也拿出来与大家分享一下吧。

12.3.3 实例应用

【例 12-2】设置单元格边界之间的距离

(1) 新建一个 border-spacing.html 文件，在该文件中首先声明两个表格存储客户信息，为这两个表格分别设置 class 属性为“one”和“two”，代码如下所示：

```
<table class="one" border="1">
    <tr>
```

```
        <td>客户姓名</td>
        <td>出生日期</td>
        <td>性别</td>
        <td>年龄</td>
    </tr>
    <tr>
        <td>江小鱼</td>
        <td>1985-02-07</td>
        <td>男</td>
        <td>25</td>
    </tr>
    <tr>
        <td>铁心兰</td>
        <td>1988-09-23</td>
        <td>女</td>
        <td>22</td>
    </tr>
</table>
<br />
<table class="two" border="1">
     <tr>
        <td>客户姓名</td>
        <td>出生日期</td>
        <td>性别</td>
        <td>年龄</td>
    </tr>
    <tr>
        <td>江小鱼</td>
        <td>1985-02-07</td>
        <td>男</td>
        <td>25</td>
    </tr>
    <tr>
        <td>铁心兰</td>
        <td>1988-09-23</td>
        <td>女</td>
        <td>22</td>
    </tr>
</table>
```

(2) 使用 border-spacing 属性来设置上述定义的两个表格的单元格边界之间的距离，代码如下所示：

```
<style type="text/css">
table.one
{
    border-collapse: separate;
    border-spacing: 10px;           //设置表格单元格水平垂直间距都为 10px
    color: #0F0;
}
```

```
table.two
{
    border-collapse: separate;
    border-spacing: 30px 20px;      //设置表格单元格水平间距为 30px，垂直间距为 20px
    color: #0F0;
}
</style>
```

12.3.4　运行结果

在 IE 浏览器中打开 border-spacing.html 文件，执行效果如图 12-3 所示。



图 12-3　设置表格单元格间距

12.3.5　实例分析



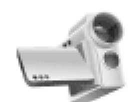
源码解析：

本实例定义了两个表格，第一个表格的 class 属性值设为“one”，第二个表格的 class 属性值设为“two”。“one”表格的 border-spacing 属性值设为 10px，表示表格单元格的水平和垂直的间距均为 10px；“two”表格的 border-spacing 属性值设为 30px 20px，表示表格单元格的水平间距为 30px，垂直间距为 20px。

12.4　设置表格标题的位置

为了帮助用户快速阅读表格，制作表格时为表格设置一个标题是非常必要的。而且同样的东西位置排放得当，也会起到很好的修饰作用，使表格看起来更加美观大方。本节将介绍一个

用于设置表格标题位置的属性——caption-side。

**视频教学：**光盘/视频/12/12.4　设置表格标题的位置.avi　　长度：3 分钟

12.4.1　基础知识——caption-side

caption-side 属性用于设置表格标题的位置，它的可选属性值如表 12-3 所示。

表 12-3　caption-side 的属性值及说明

属 性 值	说　明
top	默认值。把表格标题定位在表格之上
bottom	把表格标题定位在表格之下
inherit	规定应该从父元素继承 caption-side 属性的值

该属性的基本语法格式：

```
caption-side : top |bottom
```



有些版本的 IE 浏览器不支持 caption-side 属性，如果已规定!DOCTYPE，那么 Internet Explorer 8(以及更高版本)就可支持 caption-side 属性。

12.4.2　实例描述

一般情况下表格的标题要放在表格的上方，但在一些展示信息量比较大的网页中，就不一定了，需要考虑整体页面布局，适当地设置表格标题的位置。本节实例将为大家演示如何设置表格标题的位置。

12.4.3　实例应用

【例 12-3】设置表格标题的位置

(1) 新建一个 caption-side.html 文件，在该文件中首先定义一个员工信息表，代码如下所示：

```
<table class="one" border="1">
<caption>员工信息表</caption>
    <tr>
        <td>员工姓名</td>
        <td>出生日期</td>
        <td>性别</td>
        <td>年龄</td>
    </tr>
    <tr>
        <td>秦小兰</td>
        <td>1985-02-07</td>
```

```
        <td>女</td>
        <td>25</td>
    </tr>
    <tr>
        <td>霍斌</td>
        <td>1981-05-27</td>
        <td>男</td>
        <td>29</td>
    </tr>
</table>
```

(2) 设置 caption-side 属性来改变表格标题的位置，默认的情况下在表格的上方，下面设置表格标题处于表格的下方，代码如下所示：

```
<style type="text/css">
table.one
{
    border-collapse: separate;
    border-spacing: 10px;
    caption-side:bottom;          //设置表格的标题处于表格的下方
    color: #0F0;
}
</style>
```

12.4.4 运行结果

在 IE 浏览器中打开 caption-side.html 文件，执行效果如图 12-4 所示。

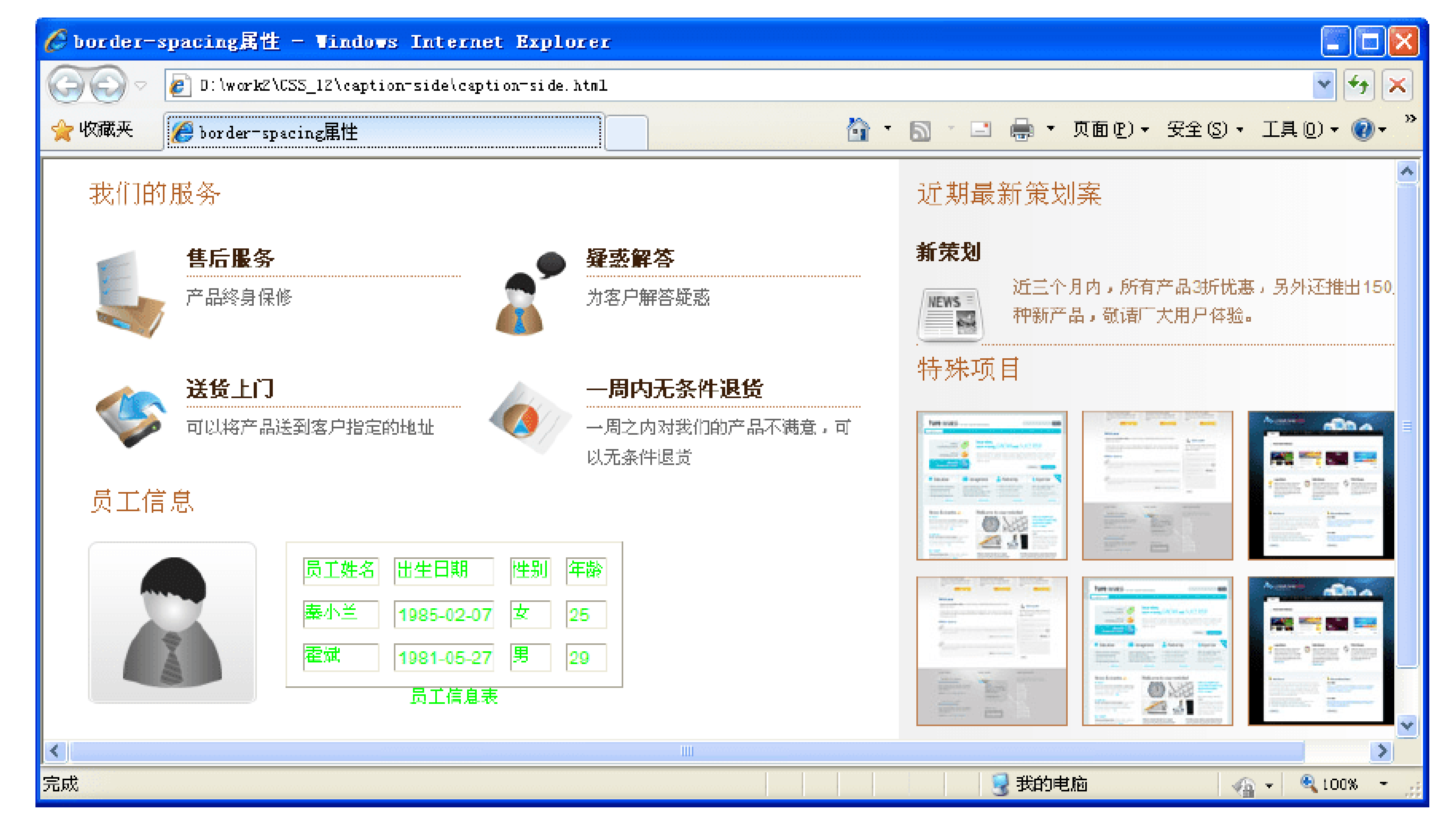


图 12-4　设置表格标题位置

12.4.5　实例分析


源码解析：

本实例中首先定义了一个员工信息表，然后使用<caption>标签为表格定义一个标题，最后设置 caption-side 属性值为 bottom，使表格标题处于表格的下方。

12.5　隐藏表格中空单元格上的边框和背景

有时表格中的一些字段不是必填内容，这样表格中就会出现空单元格的情况。如何来处理修饰这些空单元格呢？本节将介绍一个设置是否显示表格中的空单元格的属性。


视频教学：光盘/视频/12/12.5　隐藏表格中空单元格上的边框和背景.avi　长度：4 分钟

12.5.1　基础知识——empty-cells

empty-cells 属性用于设置是否显示表格中的空单元格，如果显示，那么会绘制出单元格的边框和背景。而且当 border-collapse 设置为 separate 时，此属性还有效，否则将忽略这个属性。该属性的可选属性值如表 12-4 所示。

表 12-4　empty-cells 的属性值

属 性 值	说　明
hide	默认值。不在空单元格周围绘制边框
show	在空单元格周围绘制边框
inherit	规定应该从父元素继承 empty-cells 属性的值

empty-cells 属性的基本语法格式：

```
empty-cells : show | hide
```

下面使用 empty-cells 属性来设置使表格中的空单元格显示出来，代码如下所示：

```
table.one
{
    border-collapse: separate;
    border-spacing: 10px;
    empty-cells: show;      //设置空单元格显示出来
    color: #0F0;
}
```

在 IE 浏览器中的执行效果如图 12-5 所示。



图 12-5　设置空单元格显示出来

12.5.2　实例描述

项目终于完成了，我也可以清闲一段时间了，不过这次项目中我负责的模块处理表格的比较多，还有了一些新发现，比如：隐藏表格中空单元格上的边框和背景。这些功能都可以使我们制作的页面更加完整、美观。本节实例将向大家讲解一下隐藏表格中空单元格的边框和背景的实现方法。

12.5.3　实例应用

【例 12-4】隐藏表格中空单元格上的边框和背景

(1) 新建一个 empty-cells-exp.html 文件，在该文件中定义一个存储集团信息的表格，代码如下所示：

```
<table class="tablesep" border="1" width="100%">
    <tr>
        <td>集团编码</td>
        <td>集团名称</td>
        <td>地址</td>
        <td>电话号码</td>
    </tr>
    <tr>
        <td>001</td>
        <td>汇智科技</td>
        <td>郑州市</td>
        <td>0371-66668888</td>
    </tr>
    <tr>
        <td>002</td>
        <td>天天向上有限公司</td>
        <td>重庆市</td>
        <td></td>
    </tr>
</table>
```

(2) 在该文件中使用 empty-cells 属性，设置将空单元格的边框隐藏，代码如下所示：

```
.tablesep {
    border-collapse:separate;
    empty-cells:hide;            //设置隐藏表格中空单元格的边框
    border-spacing: 10px;
}
```

12.5.4 运行结果

在 IE 浏览器中打开 empty-cells-exp.html 文件，执行效果如图 12-6 所示。



图 12-6 隐藏表格中空单元格上的边框

12.5.5 实例分析



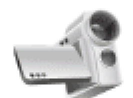
源码解析：

本实例在 empty-cells-exp.html 文件中定义一个集团信息表格，通过设置 border-collapse 属性值为 separate，使表格的边框线为分开的，empty-cells 属性值为 hide，使表格中空单元格上的边框和背景隐藏。

12.6 隐藏单元格超出的数据

控制表格的整体布局是非常重要的，一个好的布局不仅看起来美观，便于用户阅读表格信息，也使浏览器可以更快速地给用户提供信息。本节将介绍一个设置表格整体布局算法改善表

格呈递性的属性。



视频教学：光盘/视频/12/12.6　隐藏单元格超出的数据.avi　　长度：5分钟

12.6.1　基础知识——table-layout

table-layout 属性可以改善表格呈递性能。此属性导致 IE 以一次一行的方式呈递表格内容从而以更快的速度给用户提供信息。该属性的可选值如表 12-5 所示。

表 12-5　table-layout 的属性值

属 性 值	说　明
automatic	默认值。默认的自动算法，布局将基于各单元格的内容。表格在每一单元格内所有内容读取计算之后才会显示出来
fixed	固定布局的算法。在这种算法中，表格和列的宽度取决于 col 对象的宽度总和，假如没有指定，则会取决于第一行每个单元格的宽度。假如表格没有指定宽度(width)属性，则表格被呈递的默认宽度为 100%

通过 table-layout 属性值可以设置自动表格布局和固定表格布局，这两种布局格式具体介绍如下。

- 固定表格布局

  固定表格布局与自动表格布局相比，允许浏览器更快地对表格进行布局。

  在固定表格布局中，水平布局仅取决于表格宽度、列宽度、表格边框宽度、单元格间距，而与单元格的内容无关。

- 自动表格布局

  在自动表格布局中，列的宽度是由列单元格中没有折行的最宽的内容设定的。

  此算法有时会较慢，这是由于它需要在确定最终的布局之前访问表格中所有的内容。



设置此属性值为 fixed，有助于提高表格性能。对于长表格的效果尤为显著。

table-layout 属性的基本语法格式如下：

```
table-layout : auto | fixed
```

下面使用 table-layout 属性来设置表格的布局，代码如下所示：

```
<style type="text/css">
    .tablesep
    {
        table-layout: automatic
    }
    .tablecoll
    {
        table-layout: fixed
    }
</style>
```

在浏览器中运行的效果如图 12-7 所示。



图 12-7　设置表格布局



设置表格行高可以进一步提高呈递速度，浏览器不需要检测行内每个单元格的内容去确定行高，就可以开始解析以及呈递。此属性对于 currentStyle 对象而言是只读的。对于其他对象而言是可读写的。

12.6.2　实例描述

前段时间在网上看到有人问，怎样实现隐藏单元格超出的数据，当时我想了半天也不知道该怎么实现，今天突然实现了，就拿出来和大家分享一下。

12.6.3　实例应用

【例 12-5】隐藏单元格超出的数据

(1) 新建一个 table-layout-exp.html 文件，在该文件中定义一个表格记录各大主流技术的信息，代码如下所示：

```
<table class="aa" border="2" cellpadding="0" cellspacing="0"
bordercolor="#F9F9F9" id="dataLayer">
  <colgroup>
  <col width="119" />
  <col width="199" />
  <col width="99" />
  </colgroup>
  <tbody>
    <tr basestyle="oRowLine2">
      <td>J2SE</td>
      <td>Java 2 Standard Edition </td>
      <td> </td>
    </tr>
    <tr basestyle="oRowLine1">
      <td>SMS</td>
      <td>Short Message Service </td>
```

```
    <td>短信息服务</td>
  </tr>
  <tr basestyle="oRowLine2">
    <td>Windows Mobile Smartphone </td>
    <td> </td>
    <td> </td>
  </tr>
  <tr basestyle="oRowLine1">
    <td>MIDlet</td>
    <td> </td>
    <td>按 MIDP 规范开发的 J2ME 应用程序</td>
  </tr>
  <tr basestyle="oRowLine2">
    <td>MIDP</td>
    <td>Mobile Information Device Profile </td>
    <td>移动信息设备框架</td>
  </tr>
  <tr basestyle="oRowLine1">
    <td>Profile</td>
    <td> </td>
    <td>框架/简表</td>
  </tr>
  <tr basestyle="oRowLine2">
    <td>KVM</td>
    <td>K virtual Machine </td>
    <td> </td>
  </tr>
  <tr basestyle="oRowLine1">
    <td>SDK</td>
    <td>Software Development Kit </td>
    <td>软件开发工具包</td>
  </tr>
  <tr basestyle="oRowLine1">
    <td>JAD</td>
    <td>Java Application Descriptor </td>
    <td>应用程序描述符</td>
  </tr>
  <tr basestyle="oRowLine2">
    <td>GCF</td>
    <td>General Connection Framework </td>
    <td> </td>
  </tr>
  <tr basestyle="oRowLine1">
    <td>RMS</td>
    <td>Record Management System </td>
    <td>记录管理系统</td>
  </tr>
  <tr basestyle="oRowLine2">
    <td>Sprite</td>
```

```
      <td> </td>
      <td>精灵</td>
    </tr>
  </tbody>
</table>
```

(2) 设置表格的布局，使各单元格不允许换行，隐藏超出的数据，代码如下所示：

```
<style>
.aa{
    table-layout : fixed;
}
td{
    overflow:hidden;
    height:22px;
}
</style>
```

12.6.4　运行结果

在 IE 浏览器中打开 table-layout-exp.html 文件，执行效果如图 12-8 所示。



图 12-8　隐藏单元格超出的数据

12.6.5 实例分析



源码解析：

本实例在页面 table-layout-exp.html 中定义了一个记录主流技术信息的表格，该表格的 table-layout 属性设置为 fixed，使表格布局为固定模式，即表格的每列宽度固定不变，并且设置了 td 单元格的 overflow 属性值为 hidden，这样就使表格的单元格不换行，超出的数据隐藏了。

12.7 常见问题解答

12.7.1 border-collapse 问题



CSS 问题 border-collapse:collapse。

网络课堂：http://bbs.itzcn.com/thread-12394-1-1.html

我使用 border-collapse:collapse;border: 1px solid #000000 这样的设置，虽然能够将表格合并，但合并后的颜色却很浅，如何才能更换合并后表格的颜色，我试了多种办法都无效。

【解决办法】#000000 边框的颜色为黑色，要重新设置背景色的 background 属性。你感觉浅的话可能是由于 1px 造成的视觉效果吧，可以加宽边框。要更换边框颜色的话只需设置“#000000”为 RGB 颜色就可以了。

12.7.2 CSS 表格样式问题



CSS 表格样式问题。

网络课堂：http://bbs.itzcn.com/thread-12395-1-1.html

我做了一个简单的登录，前台用的是 CSS 控制样式，首先给当前的表格设置了一个 id，然后在一个框架里面用了绝对定位。代码如下所示：

```
#t{
    position:absolute;
    top:0px;
}
```

怎么表格不能紧靠顶端呢？我想让表格和顶端无缝连接，我又尝试使用以下代码：

```
body {
    text-align:top
}
```

还是不行，求高手帮忙解释。

【解决办法】使用以下代码就可以了：

```
body {
    position:relative;
}
```

12.7.3　border-collapse 取值的区别



CSS 中 border-collapse 的两个值有什么区别？

网络课堂：http://bbs.itzcn.com/thread-12396-1-1.html

border-collapse : separate | collapse，请问这两个值有什么区别？我试了一下，只不过是表格一个带边框，一个不带边框而已。我想没这么简单吧，如果是这样的话要这个属性就没用了，直接设置 table 的 border=0 或等于特定值不就行了？

【解决办法】可能是你没有仔细看效果吧，当 border-collapse 属性值设为 separate 时，边框样式是：

||　　　||　边框线是分开的。

当该属性值设为 collapse 时，边框会合并为一个单线的边框，如下：

|　　　|　边框线是合并的。

12.8　习　　题

1. 填空题

(1)　border-collapse 属性可以设置表格的边框是否合并为一个单线边框，还是在标准的 HTML 中显示一个分开边框。该属性有 3 个可选值，分别是 separate、__________和 inherit。

(2)　empty-cells 属性用于设置是否显示表格中的空单元格，下面使用该属性设置表格中空单元格是否显示，代码如下所示：

```
table.one
{
    empty-cells: show;
    color: #0F0;
}
```

上述代码设置表格__________空单元格。

(3)　设置表格单元格之间的间距，代码如下所示：

```
table
{
    border-spacing: 10px;
    color: #0F0;
}
```

其中 10px 代表__________。

2. 选择题

(1) 下面使用 border-spacing 属性设置表格边框样式，代码如下所示：

```
.tablesep
{
    border-collapse: separate;
}
```

以下对上述设置表格边框样式描述正确的是__________。

A. 表格边框宽度为 separate　　B. 表格边框为分开样式

C. 表格边框合并为一个单线样式　　D. 以上选项都不正确

(2) 下面__________属性可以用来指定分隔边框模型中单元格边界之间的距离。

A. border-collapse　　B. border-spacing

C. caption-side　　D. empty-cells

(3) caption-side 属性设置表格标题的位置，代码如下所示：

```
table.one
{
    caption-side:bottom;
}
```

该代码设置了表格标题的位置为__________。

A. 在表格的上方　　B. 在表格的右方

C. 在表格的左方　　D. 在表格的下方

(4) 下面可以设置表格的布局的属性是__________。

A. caption-side 属性　　B. empty-cells 属性

C. table-layout 属性　　D. border-spacing 属性

3. 上机练习

上机练习：制作一个客户信息表

练习要求：客户信息表需要包含客户名称、性别、年龄和交易日期。表格定义完成之后，需要设置表格单元格水平与垂直距离都为 15px，表格的标题位于表格的下方，空的单元格也要显示周围边框，并且设置表格为固定布局。执行效果如图 12-9 所示。

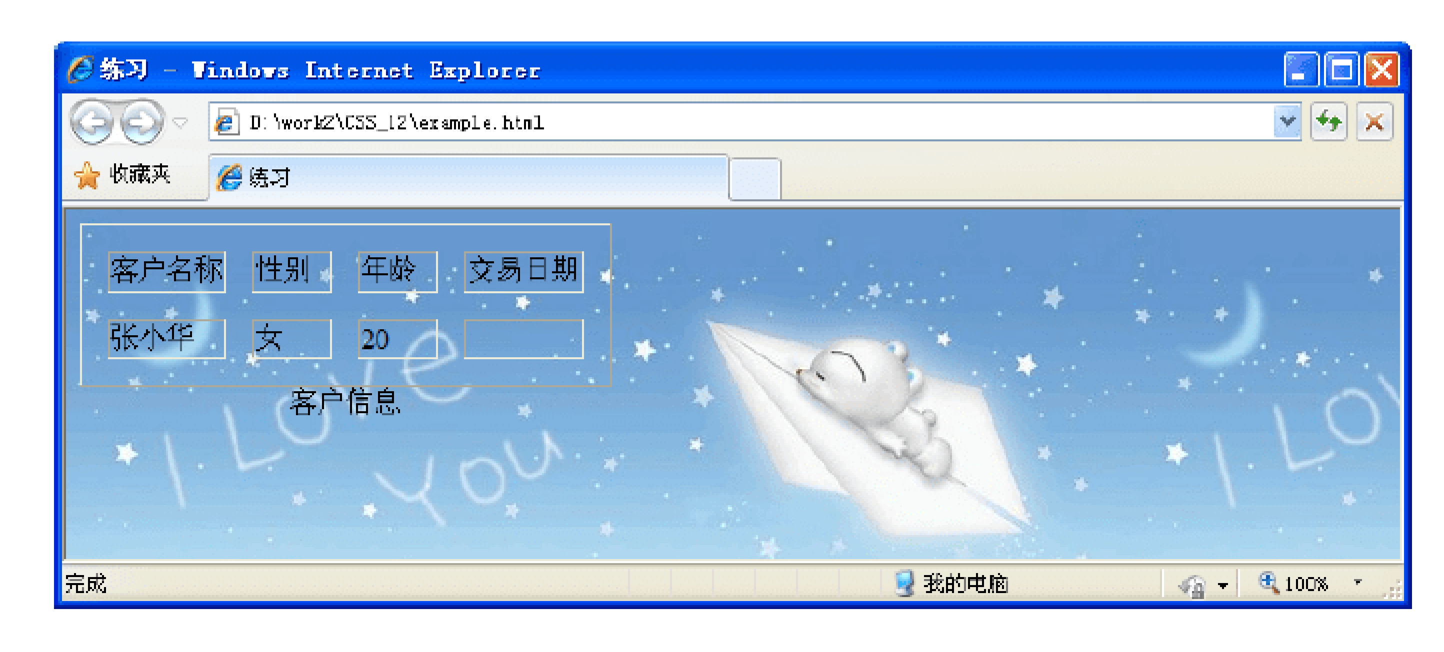


图 12-9　客户信息表



第 13 章　逐步改善页面的表单

内容摘要：

表单是网站设计和开发必不可少的一部分，缺了它，我们很多事情都无法完成。例如：向用户获取个人数据，向公告栏提交信息，为购物车添加物品，更新微博信息等。

虽然表单在网页中十分重要，但是非常遗憾的是：HTML 并没有为设计者提供任何可选的样式，所以表单只能以浏览器默认的样式显示。为了弥补这些不足，CSS 可谓是雪中送炭，提供了许多在文档中定位表单元素的方法，使表单元素能够使用 CSS 样式规则。

本章将介绍一些相对生僻的、能增强表单可用性的 HTML 标签、属性，及一些额外可用的 CSS 元素。但是值得我们注意的是：不能一味地注重表单的外形美观，也要保证表单的可用性。因为表单是用来获取用户输入信息的，但是如果用户都不知道应该如何使用他们，表单再漂亮也没有任何意义。

学习目标：

- 制作特殊样式的文本框
- 同一个表单中制作特殊样式的文本框和按钮
- 创建与文档同行的搜索框
- 对相关的字段进行分组
- 为快捷键提示信息设置样式
- 制作不同背景的下拉菜单
- 强调显示当前的焦点字段

13.1 制作特殊样式的文本框

HTML 表单元素是用来输入用户信息和提交数据的。由于 HTML 没有提供任何可选的样式，所以我们想要设置“文本框”、“下拉列表框”等表单元素的样式就得费点心思了。CSS 为我们提供了许多在文档中定位表单元素的方法，本节我们来研究一下 CSS 如何为表单元素设置样式。

**视频教学：** 光盘/视频/13/13.1 制作特殊样式的文本框.avi 长度：4 分钟

13.1.1 基础知识——CSS 为表单元素设置样式

如果需要改变 form、input 等表单元素的外观，我们可以通过 CSS 为 form、input、textarea、select 等 HTML 元素设置样式来修改这个外观。

例如：

```
input{
        color:#red;
        background-color:#B898F3;
        border:1px solid #00008b;
        font:0.9em Arial , Helvetica,sans-serif;
        padding:0.2em; width:200px;
}
```

下面我们通过表 13-1 来分析这些样式。

表 13-1 input 中的属性及其说明

属 性	说 明
color	改变字段中输入的文本的颜色
background-color	定义字段的背景颜色
border	改变字段的边框，所有其他边框的样式都可以使用
font	改变字段中字体的大小和颜色
padding	设置输入框和字段中输入的文字之间的间隔
width	允许用户输入的最大宽度(例如不希望用户输入太长的内容)



form、input、select、textarea 都有表 13-1 所示的那些属性，同样地通过修改这些属性，可以改变 select、textarea 等表单元素的外观。

13.1.2 实例描述

我们在注册用户信息和使用用户名及密码登录时，或者是利用搜索引擎搜索信息时，都一

定会用到文本框。而文本框的样式也是品种繁多。为了看起来显得美观大方，再小的细节也必须要高度重视。例如：文本框的边框样式，背景颜色，输入内容距离文本框的间隙等，这都会影响页面的整体效果。

下面就让我们来做一个有特殊样式的输入框的登录窗口。

13.1.3　实例应用

【例 13-1】制作特殊样式的文本框

(1) 创建一个名为 login.html 的页面。

(2) 为表单元素定义样式，代码如下：

```
form {
    padding:0px;
}
input {
    background-color:#E8EcF8;
    color:red;
    border:thin solid #798DB9;
    font:0.9em Arial , Helvetica,sans-serif;
    padding:0.2em;
    height:12px;
}
```

(3) 在 body 元素中表单元素的代码如下所示：

```
<form class="bg">
  <div id="apDiv1">
    <input type="text" />
    <div id="apDivtab"></div>
    <input type="text" />
  </div>
</form>
```

(4) 为了使界面看起来更加美观，我们布局一下页面，效果如图 13-1 所示。



图 13-1　布局好的页面效果

(5) 保存修改的代码。

13.1.4 运行结果

直接打开浏览器，界面运行的效果如图 13-2 所示。



图 13-2 设置好的背景色和边框样式的效果

设置字体颜色为红色，显示的效果如图 13-3 所示。



图 13-3 input 元素中字体为红色的效果

13.1.5 实例分析



源码解析：

在本章实例中，定义了 input{}表单元素的样式，可以看到该页面中所有 input 元素都会使用已经定义好的 input 样式。因此需要什么样的效果直接在某个表单元素中设置样式即可。

13.2 制作特殊样式的文本框和按钮

通过上节的学习，我们知道 input 元素有很多种类型。而对于文本框应用的样式和对于按钮、单选按钮、复选按钮应用的样式截然不同。如何为不同的表单字段创建特定的样式呢？

本节将介绍如何为同一个表单中的元素应用不同的样式。

**视频教学：**光盘/视频/13/13.2　制作特殊样式的文本框和按钮.avi　　长度：4 分钟

13.2.1　基础知识——同一个表单中的元素应用不同的样式

使用 CSS 类可以单独为某种字段指定样式规则。但如果是不同种类的字段呢，例如：有些文本框可能只需要输入两个字符，而有些则可能需要输入一个名称或较短的单词，还有一些可能必须输入一个完整的句子。那我们就需要为这些较小、中等、较大的文本框创建 CSS 类，让它们能够选择与用户输入数据相适应的样式。这样就能帮助用户确认它们输入的信息是否合适。

下面我们以一个显示为文本框，一个显示为 Submit(提交)按钮为例做一下对比。为所有输入元素创建一组相同的样式(特定的宽度加上一个内陷的边框)，代码如下所示：

```
input {
    background-color:#e1f4ee;
    border:groove 1px #C6EAE8;
    color:red;
    width:120px;
}
```

显示的结果如图 13-4 所示。



图 13-4　将同一样式应用到不同的字段中

图 13-4 中【确定】和【取消】按钮看起来更像一个文本框，而不像一个按钮。为了弥补这样的缺陷，我们需要为同一表单中不同种类的字段设置特定的样式。代码如下：

```
input.txt {
    background-color:#F60;
    color:red;
    width:120px;
```

```
    border:solid 1px #C6EAE8;
}
input.btn {
    background-color:#FFF;
    border:solid 1px #DBDBDB;
    width:80px;
}
```



提到按钮，我们不得不注意按钮的 background-color 属性，当它的属性设置为 Transparent 时，按钮的显示情况和文本没有什么分别。

13.2.2 实例描述

如果一个表单中有“输入框”和“按钮”，当我们为所有 input 元素设置一组相同的样式时，浏览器显示的效果如图 13-4 所示。不是很美观，所以我们要设置不同的类，使每个表单元素呈现不同的效果。

13.2.3 实例应用

【例 13-2】同一个表单中制作特殊样式的文本框和按钮

(1) 创建一个 loginbtn.html 页面。

(2) 为页面定义一组样式，代码如下：

```
input.txt {
    background-color:#F60;
    color:red;
    width:120px;
    border:solid 1px #C6EAE8;
}
input.btn {
    background-color:#FFF;
    border:solid 1px #DBDBDB;
    width:80px;
}
```

(3) 在<body>…</body>中添加如下代码：

```
<form class="bg">
  <div id="apDiv1">
    <input  type="text" class="txt"/>
    <div id="divtab"></div>
    <input type="text" o class="txt"/>
  </div>
  <div id="apDivbtn">
    <input type="button" value="确定" class="btn" />
```

```
    <input type="button" value="取消" class="btn"/>
  </div>
</form>
```

(4) body 元素里面为 id 设置的样式代码本例中不做解释。

(5) 保存修改的代码。

13.2.4　运行结果

直接打开浏览器运行上述代码，显示的效果如图 13-5 所示。



图 13-5　对“输入框”和“按钮”应用不同样式的效果

13.2.5　实例分析



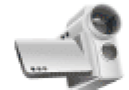
源码解析：

我们以前学过类样式表，在 style 元素中定义一个类样式，在用到此样式时，通过 class 调用。现在我们为不同类型的元素设置样式与类样式表颇为相似，都需要用 class 来调用，只是在类样式表的前面需要加上表单元素。这样表单就适应多种不同类型的数据类型了。

13.3　创建与文档同行的搜索框

表单和段落相似，默认的情况下将以新行显示，这一点很受我们欢迎。但有时我们希望在一个标题之后添加一个表单，例如：某购物网站头部的右上角会显示一个搜索框。

下面我们就来看一下这样的效果是如何实现的。



视频教学：光盘/视频/13/13.3　创建与文档同行的搜索框.avi　　　　长度：2 分钟

13.3.1 基础知识——避免表单创建额外的空格和换行符

在 div 元素中，如果想让内容在同一行显示，只需要将 div 的 display 属性值设置为 inline 就可以了。当然对于表单来说，我们也可以设置它的 display 属性值。

例如：

```
<style type="text/css">
form {
    display:inline;
}
</style>
```

13.3.2 实例描述

在浏览购物网站时，为了使用方便，我们总是在右上角的搜索框里搜索商品。同时我们也注意到：这些网站上的搜索框都是紧挨着图片或者文档的，看起来让人赏心悦目。但是如果再换行显示，不太符合页面的整体效果。

下面将介绍一个很简单的例子，让我们来研究一下设置 display 属性和不设置 display 属性的区别。

13.3.3 实例应用

【例 13-3】创建与文档同行的搜索框

(1) 创建一个名为 inline.html 的页面。

(2) 为 inline.html 中的 form 表单定义样式，代码如下：

```
<style>
form{
    display:inline;}
form input {
    border:1px solid #FFF;
    background-color:#FFF;
}
.bg {
    background-image:url(img/inline.jpg);
    background-repeat:no-repeat;
    width:803px;
    height:431px;
}
#apDiv1 {
    position:relative;
    width:798px;
    height:38px;
```

```
    padding-top:30px;
}
</style>
```

(3) 在 body 元素中添加如下代码：

```
<div id="apDiv1"> <font color="#FFFFFF">You are not Logged in! Log in to check
your messages. 键入关键字搜索你想要的文章：</font>
  <form>
    <input  type="text"/>
    <input type="submit"  value="搜索"/>
  </form>
</div>
```

(4) 保存修改的代码，查看运行结果。

13.3.4　运行结果

使用浏览器直接打开文件名为 inline.html 的页面，运行的结果如图 13-6 所示。

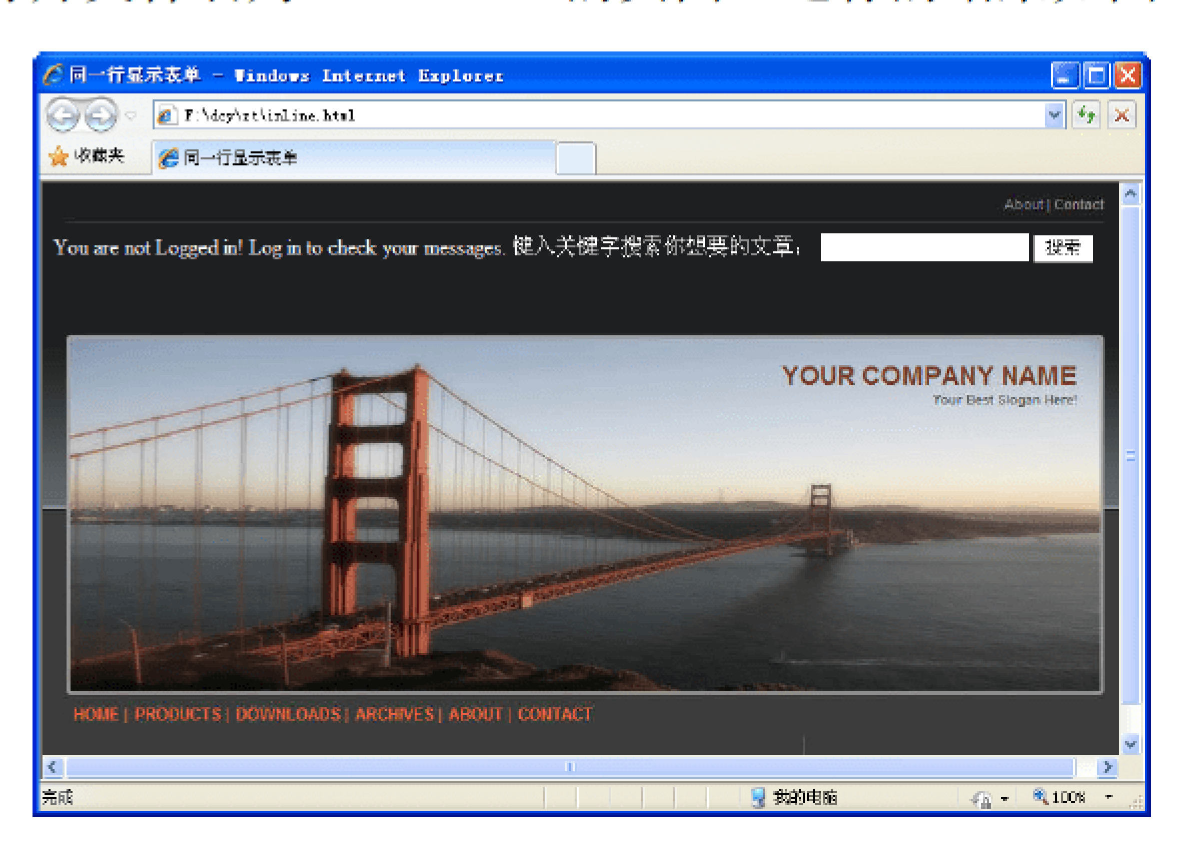


图 13-6　设置 display 属性值为 inline 显示的效果

如果我们不设置 display 属性，运行的结果如图 13-7 所示。

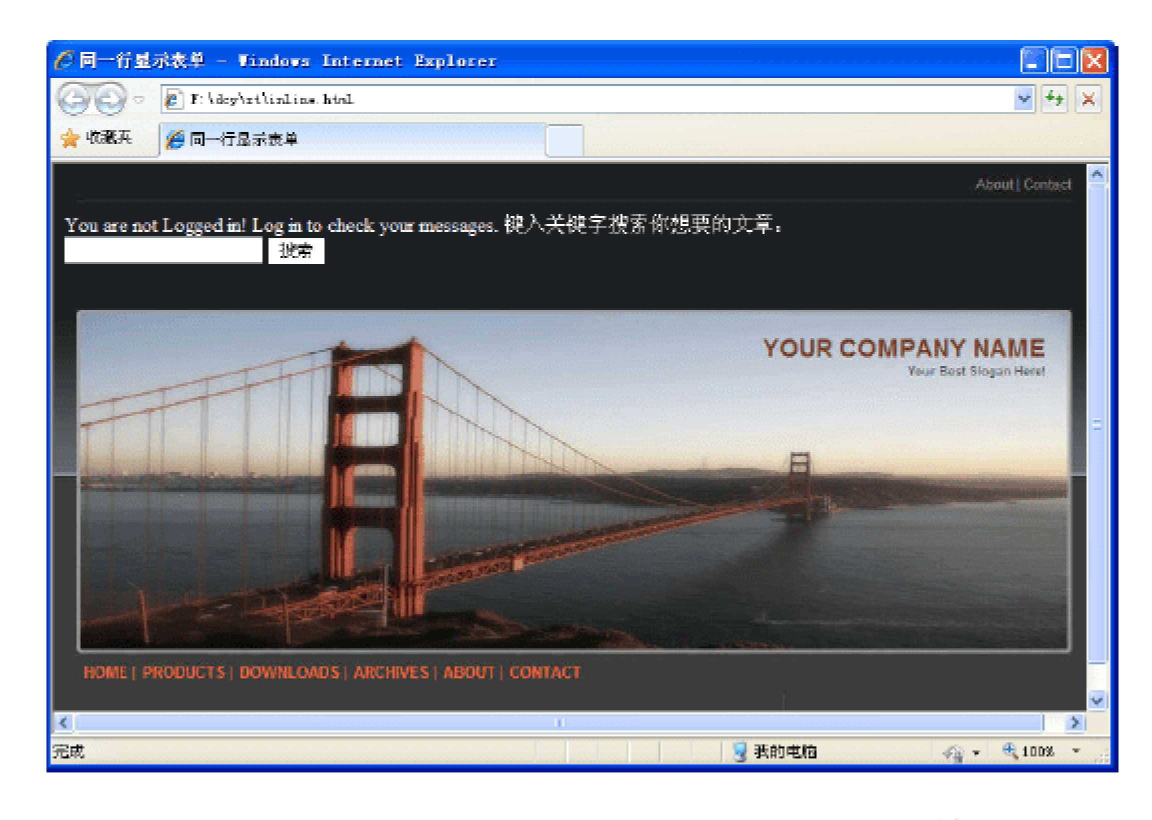


图 13-7　没有设置 display 属性显示的效果

13.3.5 实例分析


源码解析：

当我们解决问题没有头绪时，往往会把问题看得很难，但是有的问题只需要设置一个属性，就会柳暗花明。如果我们没设置 form 表单的 display 属性，是不是就需要借用 div 元素，然后将 form 元素嵌套在里面，才能实现与文档同行的搜索框。所以，设置 form 表单的 display 属性既可使代码简洁又避免了表单创建额外的空格和换行符。

13.4 对相关的字段进行分组

我们在填写简历时，会发现在第一步“基本信息”中的“联系方式”包含了电子信箱、个人主页、邮政编码、通信地址等信息。对于这个包含大量字段的表单而言使用了分区显示，更好地展现了信息间的关系，使访问者能够了解到哪些问题是有关联的。那么我们用什么方法来实现类似的效果呢？

下面我们来介绍如何对相关字段进行分组。


视频教学：光盘/视频/13/13.4 对相关的字段进行分组.av　　长度：3 分钟

13.4.1 基础知识——fieldset 元素

<fieldset>和<legend>标签可以将表单中相关联的信息分成不同的组，而它的使用方法很简单，只需简单地把相关的字段放在同一个<fieldset>标签中，并在<fieldset>开始标签后放置一个用来说明分组标题的<legend>标签。

13.4.2 实例描述

有一个登录窗口，在上面除了需要显示登录的基本信息之外，还需要注意：如果用户忘记密码或者看不清验证码，应该怎么办呢？在这种情况下，对字段进行分组无疑是最好的选择。

下面我们就来学习一个这样的例子。

13.4.3 实例应用

【例 13-4】对相关的字段进行分组

(1) 创建一个名称为 fieldset.html 的页面。

(2) 为说明分组的标题<legend>标签设置样式，其代码如下：

```
<style>
form div {
    width:219px;
}
form fieldset {
    width:219px;
}
form legend {
    color:#999;
    font-size:12px;
}
</style>
```

(3) 在<body>标签中填入如下代码：

```
<form>
  <fieldset>
    <legend>基本信息</legend>
    <div><img src="img/fieldset.jpg" /></div>
  </fieldset>
  <fieldset>
    <legend>其他信息</legend>
    <div><img src="img/fieldset2.jpg" /></div>
  </fieldset>
</form>
```

(4) 保存修改的代码，查看其运行的结果。

13.4.4　运行结果

使用浏览器直接打开名为 fieldset.html 的页面，效果如图 13-8 所示。



图 13-8　使用<fieldset>标签分组的效果

13.4.5　实例分析



源码解析：

通过该实例可以看出，<fieldset>标签的用法其实很简单，只需将想要显示的内容放在<fieldset>标签中即可，然后再使用<legend>标签说明一下分组的标题，这样一个完整的分组就实现了。

13.5　为快捷键提示信息设置样式

我们通常都非常喜欢使用快捷键，因为纯键盘的操作会在一定程度上提高工作效率；特别对于那些应用鼠标不太灵活甚至无法使用鼠标的用户而言是很有帮助的。对于我们而言，通过快捷键可以快速地定位文档某个内容或者访问一个连接，而所要做的工作仅仅是同时按下Alt(或者类似的键)和其他特定键。但我们必须先要让用户知道这个特定键是什么。

下面我们就来看一下如何设置这个特定键。



视频教学：光盘/视频/13/13.5　为快捷键提示信息设置样式.avi　　长度：5 分钟

13.5.1　基础知识——accesskey 属性

在操作系统中，通常在快捷键对应的字母下面添加一个下划线。例如在 Windows 中同时按下 Alt+F 组合键就能打开“文件”下拉菜单，而这是通过在 File 中的字母“F”的下面添加下划线来提示用户的，如图 13-9 所示。

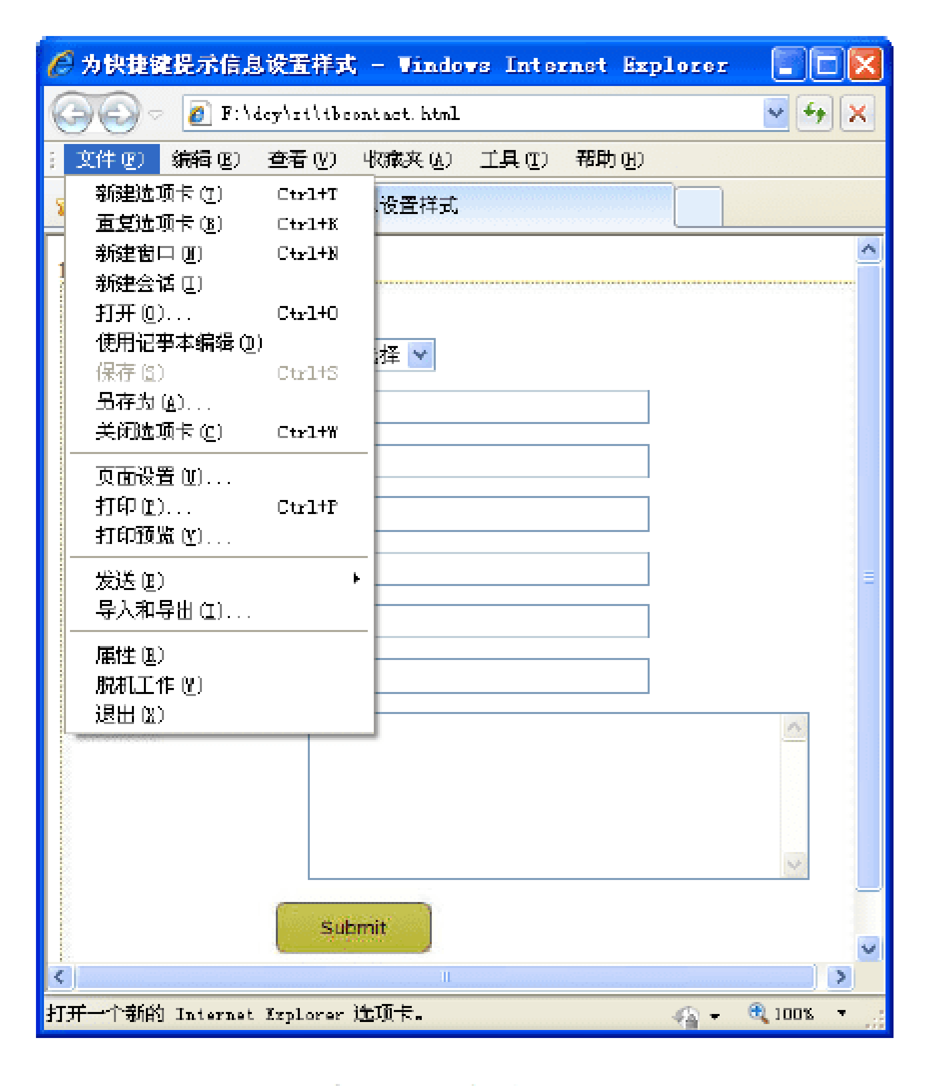


图 13-9　快捷键对应字母下面有下划线

我们在网站中也可以使用类似的方法，在作为快捷键的字符下面添加下划线。代码如下所示：

```
<style>
.akey{
    text-decoration:underline;}
</style>
```

接着为快捷键选择字符，代码如下：

```
<span class="akey">p</span>
```

要想实现表单的快捷键功能，只需通过 accesskey 属性设置我们选择作为快捷键的字符即可。

13.5.2　实例描述

对于一个初学电脑，对操作鼠标不太灵活甚至不懂的用户来说，使用键盘操作无疑是最好的方法。注册用户信息时，我们可以使用户通过按一个键的方法直接跳转到表单上，然后按另一个键直接跳转到表单的第一个元素中。这对使用这种功能的用户来说是不会陌生的。

下面的例子就很好地说明了这一点。

13.5.3　实例应用

【例 13-5】为快捷键提示信息设置样式

(1) 创建一个名为 tabcontact.html 的页面。

(2) 为表示的快捷键定义一个样式，代码如下：

```
<style>
    form input{
        width:200px;
        height:15px;}
.akey{
    text-decoration:underline;
    color:#666;
    }
</style>
```

(3) 以上样式仅给出了主要的样式代码，如何布局在这里不再详细讲解。

(4) 接着在<body>标签中添加代码，如下所示：

```
<form class="bg">
<div id="apDiv1"> <select accesskey="p">
<input type="text" accesskey="p"/>
 <textarea cols="50" rows="10" accesskey="p"></textarea>
</div>
</select>
```

(5) 以上代码主要是设置<input>标签里面的属性 accesskey 值。

(6) 保存修改后的代码。

13.5.4 运行结果

使用浏览器直接打开文件名为 tbcontact.html 的页面，看到的效果如图 13-10 所示。

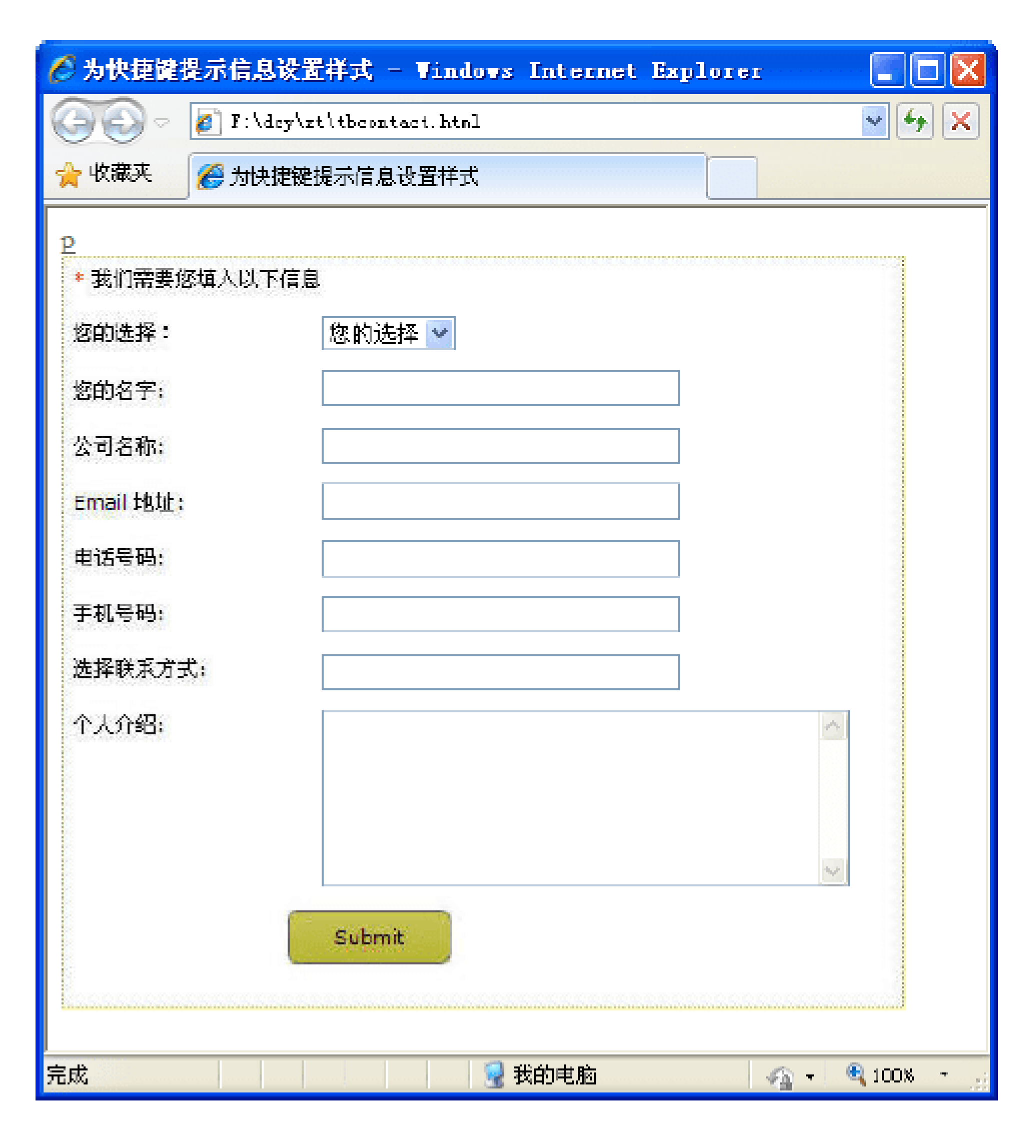


图 13-10 使用浏览器打开后看到的效果图

按下 Alt+P 组合键，效果如图 13-11 所示。

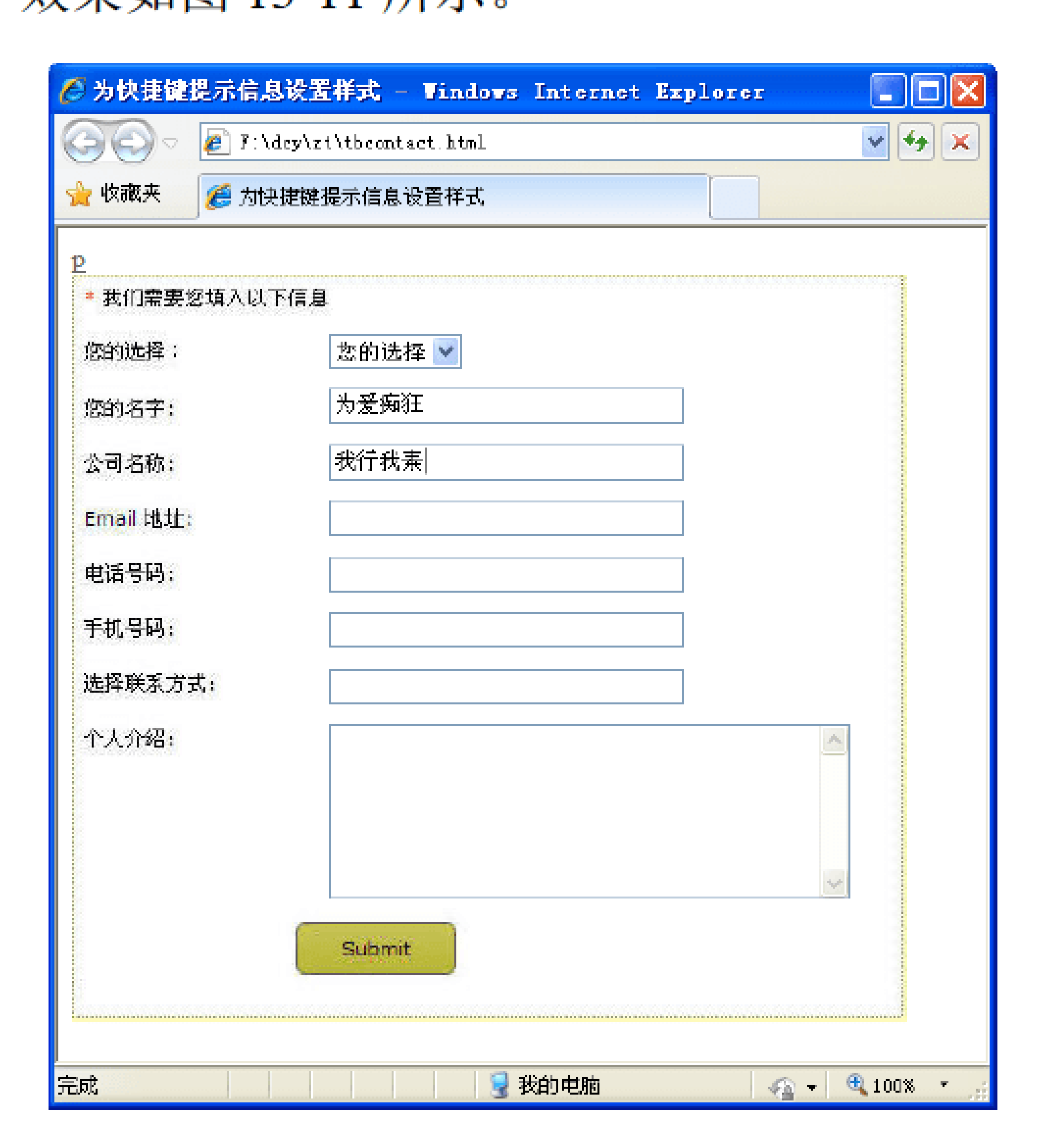


图 13-11 焦点落在表单第三个元素上的效果

再次按下 Alt+P 组合键，效果如图 13-12 所示。

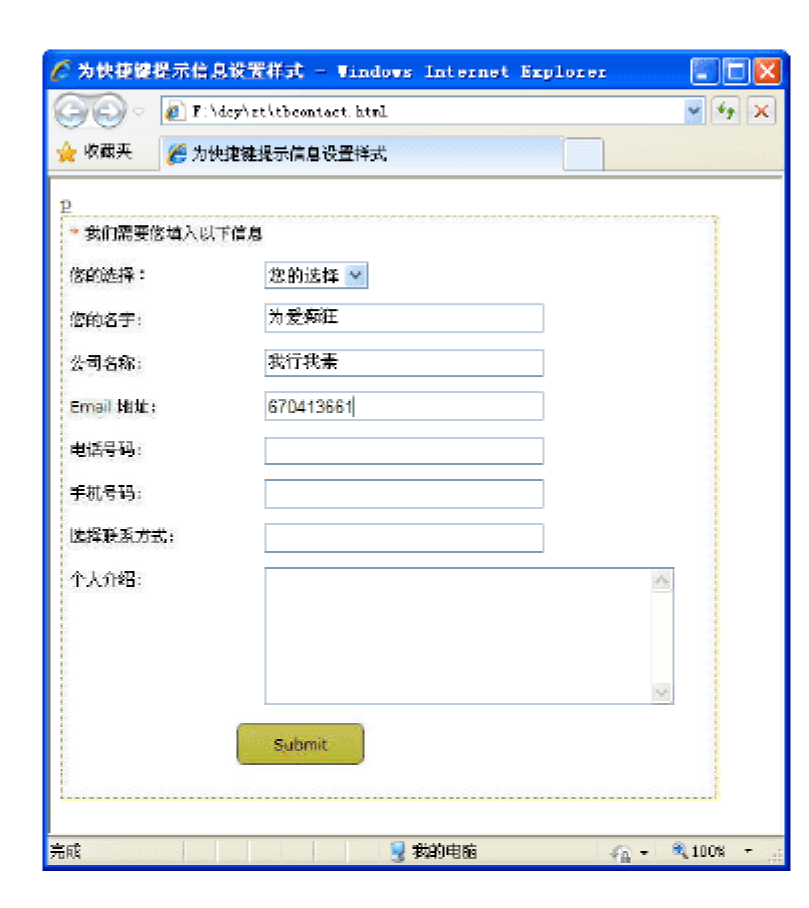


图 13-12　焦点落到表单的第四个元素上的效果

13.5.5　实例分析



源码解析：

在 IE 浏览器中是使用 Alt 键作为快捷键，而在 Firefox 浏览器中则是使用 Alt+Shift 组合键。所以并非所有的浏览器都是以相同的方式支持快捷键的。但是都允许用户配置自己的快捷键组合。

13.6　制作不同背景的下拉菜单

在前面的章节中，我们学习了为 select 菜单设置背景颜色的方法，但如何使用不同颜色来显示菜单中的不同选项呢？

下面我们就来学习，如何使用 CSS 为不同的选项设置不同的背景色。



视频教学：光盘/视频/13/13.6　制作不同背景的下拉菜单.avi　　长度：4 分钟

13.6.1　基础知识——使用不同背景强调 select 下拉菜单

在以往的菜单中，我们只是为菜单添加背景色，却没试过为不同的选项设置不同的背景色。那么用什么方法来实现这种效果呢？

因为 color 和 background-color 属性是菜单中唯一可自定义的属性，所以我们可以为各个菜单项指定不同的类，来对其应用多种不同的背景颜色。

下面我们来了解一下是如何设置的，代码如下：

```
.redcss{
        background-color:#F00;
```

```
        color:#FFF;}
    .greencss{
        background-color:green;
        color:#000;}
    .huangcss{
        background-color:#FF0;
        color:#06F;}
    .bluecss{
        background-color:blue;
        color:#0F0;}
```

13.6.2　实例描述

为不同的下拉菜单设置不同的背景色的确不经常使用，但是也不排除使用的可能，所以为了那万分之一的可能，来学习一下也是一个很不错的选择。

下面我们来看一下 select 菜单中所显示的不同颜色。

13.6.3　实例应用

【例 13-6】制作不同背景的下拉菜单

(1) 创建一个名为 select.html 的页面。

(2) 为 select 菜单定义不同的背景色，代码如下：

```
<style>
    .redcss{
        background-color:#F00;
        color:#FFF;}
    .greencss{
        background-color:green;
        color:#000;}
    .huangcss{
        background-color:#FF0;
        color:#06F;}
    .bluecss{
        background-color:blue;
        color:#0F0;}
</style>
```

(3) 页面布局，在这里就不给出样式了。

(4) 在<body>标签中添加页面代码，如下所示：

```
<form class="bg">
  <div id="apDiv1">
    <select>
      <option  selected="selected" class="redcss" >网站管理</option>
      <option value="domain" class="huangcss">域名管理</option>
```

```
      <option value="mail" class="greencss">邮局管理</option>
      <option value="database" class="bluecss">数据库管理</option>
    </select>
  </div>
</form>
```

(5) 保存修改后的代码。

13.6.4　运行结果

打开浏览器，在浏览器中运行保存后的 select.html 文件，效果如图 13-13 和图 13-14 所示。



图 13-13　背景颜色效果图



图 13-14　不同的背景色效果

13.6.5　实例分析



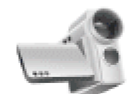
源码解析：

如果要为 select 菜单中的不同选项设置不同的字体，这样的效果就不能实现。因为 color 和 background-color 属性是菜单中唯一可自定义的属性。所以只有字体的颜色和背景色可以定义，其他的都不行。

13.7　强调显示当前焦点字段

在 Excel 表格中，当我们单击某个字段或者用 Tab 键切换时将强调显示当前的焦点字段，那么我们在 Web 中能否用 CSS 实现类似的效果呢？

下面我们来看一下如何用 CSS 强调用户单击的表单字段。


视频教学：光盘/视频/13/13.7 强调显示当前焦点字段.avi 长度：4 分钟

13.7.1 基础知识——强调用户单击的表单字段

前面我们学习了 CSS 伪类，那么这里通过:focus 伪类可以实现强调显示用户单击的表单字段的效果。



这种效果在 IE8 浏览器中可以实现，但对于应用更加广泛的 IE6 浏览器而言是无法实现的。

我们以 IE8 浏览器中显示的效果为例，为表单字段定义样式，代码如下：

```
<style type=text/css>
form input{
    border:2px solid #FFF;
    }
    form input:focus{
        border:2px solid #00F;
        background-color:#F00;
        }
</style>
```

正如我们上面提到的，IE6 浏览器的应用范围更加广泛，但 IE6 浏览器并不支持 focus 伪类，所以对于大多数浏览器用户而言是看不到这样的效果的。

13.7.2 实例描述

我们在注册或者登录时，会习惯性地按下 Tab 键，来使表单中的下一个字段获得焦点。为了分辨此字段是否选中，通常会改变选中的表单字段的样式。

我们来看一下这个例子。

13.7.3 实例应用

【例 13-7】强调显示当前焦点字段

(1) 创建一个 tabfocus.html 的页面。

(2) 在<style>标签中添加样式，代码如下所示：

```
<style>
.bg {
    background-image:url(13-1.JPG);
    background-repeat:no-repeat;
    vertical-align:middle;
    background-position:center;
```

```
    width:781px;
    height:171px;
}
#apDiv1 {
    position:relative;
    width:205px;
    height:20px;
    padding-left:210px;
    padding-top:130px;
}
form input{
    border:2px solid #FFF;
    }
    form input:focus{
        border:2px solid #00F;
        background-color:#F00;
        }
</style>
```

(3) 在<body>标签中添加的代码如下所示：

```
<form  class="bg">
<div id="apDiv1">
   <input type="text"  style="width:65px;"/>
             
    <input type="text"  style="width:65px;"/>
 </div>
 </form>
```

(4) 保存修改后的代码。

13.7.4　运行结果

.在 IE8 浏览器中打开文件名为 tabfocus.html 的页面，初始状态的效果如图 13-15 所示。

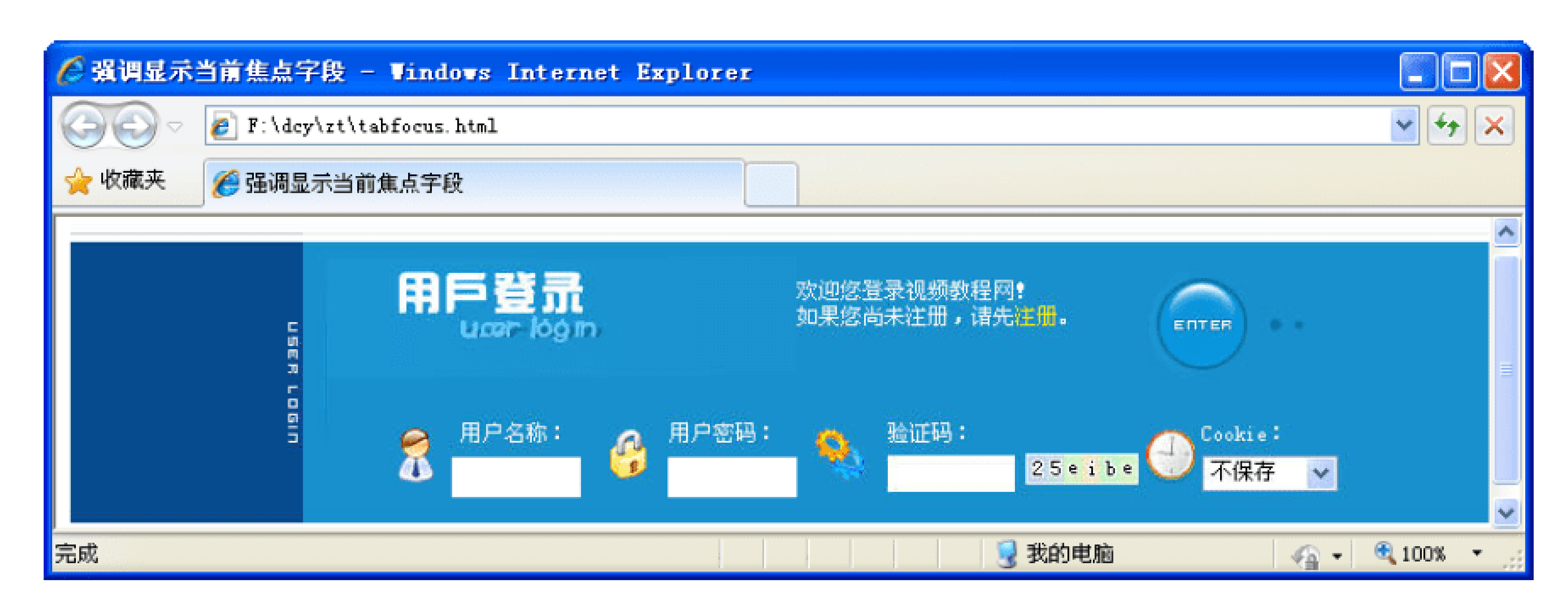


图 13-15　默认的效果

单击“用户名称”输入框时，效果如图 13-16 所示。

图 13-16　“用户名称”的输入框获得焦点时的效果

13.7.5　实例分析



源码解析：

对于:focus 伪类，IE8 浏览器能够完全支持，但 IE6 浏览器却不支持。所以，我们在以后碰到不显示效果的情况时，不要盲目地从代码中查找错误，首先要搞清楚浏览器是否支持。

13.8　常见问题解答

13.8.1　HTML 中的 fieldset 的问题



HTML 中 fieldset 的问题。

网络课堂：http://bbs.itzcn.com/thread-11630-1-1.html

对于这样一段代码：

```
<form action="dreamdu.php" method="post" id="dreamduform">
 <fieldset>
  <legend>用户名与密码:</legend>
  <input id="hiddenField" name="hiddenField" type="hidden" value="hiddenvalue" />
  <label for="username">用户名:</label>
  <input type="text" id="username" name="username" value="dreamdu" />
 </fieldset>
 <fieldset>
  <legend>性别:</legend>
  <label for="boy">男</label>
  <input type="radio" value="1" id="sex" name="sex" />
  <label for="girl">女</label>
  <input type="radio" value="3" id="sex" name="sex" />
 </fieldset>
</form>
</body>
```

我怎样才能将两个 fieldset 之间空出一两行的距离呢？

【解决办法】要解决这个问题，有一个很简单的办法，完全不需要再嵌套表格，而只需要在 style 元素里面定义 div 元素的高度即可，代码如下：

```
#divTab {
    height:8px;
}
```

然后在</fieldset>和<fieldset>标签中添加<div id="divTab"></div>，代码如下：

```
</fieldset>
<div id="divTab"></div>
 <fieldset>
```

这样，你想让两个 fieldset 之间空出几行就能空出几行了。

13.8.2　如何使用 accesskey 的问题



在 HTML 中如何使用 accesskey 的问题。

网络课堂：http://bbs.itzcn.com/thread-11631-1-1.html

表单元素中有个<label></label>表单元素，听说是用来设置表单元素快捷键的，哪位高手能给俺指点一下啊？

【解决方法】我们以 IE8 浏览器为准。

```
<label>
标识有两个属性，一个是 FOR，一个是 ACCESSKEY。FOR 的意思是，这个 Label 是为哪个元件服务的？
而 ACCESSKEY 则定义了访问这个元件的热键。比如，有一个名字叫 Name 的单行输入框，我们是这样
定义这个元件的：
<input type="TEXT" id="Name" size=30>
，那么，我们的
<label>
可以如此定义：
<label for="Name"  accesskey="N">姓  名(<U>N</U>): </label>
。大家可以看见，这个
<label> 是为那个 ID 为“Name”的元件服务的，而 ACESSKEY 则定义了访问这个元件的热键为
“Alt+N”。</label>
```

实例代码如下：

```
<FORM>
  <LABEL FOR="Namebox" ACCESSKEY="N">姓  名(<U>N</U>): </LABEL>
  <INPUT TYPE="TEXT" ID="Namebox" SIZE=30>
  <br>
  <LABEL FOR="Emailbox" ACCESSKEY="E">电子邮件(<U>E</U>): </LABEL>
  <INPUT TYPE="TEXT" ID="Emailbox" SIZE=30>
  <br>
```

```
  <LABEL FOR="Phonebox" ACCESSKEY="P">联系电话(<U>P</U>): </LABEL>
  <INPUT TYPE="TEXT" ID="Phonebox" SIZE=30>
  <br>
  <LABEL FOR="areabox" ACCESSKEY="S" >留  言(<U>S</U>): </LABEL>
  <TEXTAREA ID="areabox" COLS="30" ROWS="3"></TEXTAREA>
  <br>
  <input type="CHECKBOX" id="check3" value="often" name="checkoften">
  <label for="check3" ACCESSKEY="H">经常来这里(<U>H</U>)</label>
  <input type="CHECKBOX" id="check4" value="seldom" name="checkseldom">
  <label for="check4" ACCESSKEY="L">偶尔来看看(<U>L</U>)</label>
</FORM>
```

保存以上代码。查看运行效果。

13.9 习　　题

1. 填空题

(1) 在 form 表单中，__________属性用于设置表单和文档在同一行显示。

(2) 为表单中的相关的字段进行分组，我们需要使用__________元素。

(3) 想要实现表单的快捷键功能，只需为__________属性设置我们选择作为快捷键的字符即可。

(4) 在为 form 表单中的下拉菜单的内容设置不同的样式时，color 和__________属性是菜单中唯一可自定义的属性。

(5) 假设有一个文本框，代码如下所示：

```
<input  type="text" />
```

当这个文本框获得焦点时，使文本框的边框和背景颜色为蓝色，字体为白色，应该使用的代码是__________。

2. 选择题

(1) 假设一个表单中有一个文本框和一个按钮，分别为它们设置不同的样式，调用的代码如下所示：

```
<input  type="text" class="TXT" />
<input  type="Submit" class="BTN"  />
```

为此表单设置的样式正确的是__________。

A.

```
input: TXT{border:thin solid #00F;}
input: BTN{background-color:#00F; color:#FFF;}
```

B.

```
input  TXT{border:thin solid #00F;}
input  BTN{background-color:#00F;color:#FFF;}
```

C.

```
input  .TXT{border:thin solid #00F;}
input  .BTN{background-color:#00F;color:#FFF;}
```

D.

```
input.TXT{border:thin solid #00F;}
input.BTN{background-color:#00F;color:#FFF;}
```

(2) 在 form 表单中，__________用于设置表单和文档在同一行显示。

A. display: inline　　　B. display: inline-block

C. display: inline-table　　　D. display: block

(3) 下面__________项是对相关的表单字段进行分组的。

A. legend　　B. accesskey　　C. fieldset　　D. span

(4) 假设要设置一个类似文本的按钮，正确的是__________。

A. background:transparent; border: 0; padding: 0;

B. background-color: transparent; border: 0; padding: 0;

C. background: white; border: 0; padding: 0;

D. backgorund-color: white; border: 0; padding: 0;

3. 上机练习

上机练习：制作特色的登录窗口

通过本章的学习，我们对使用 CSS 为表单元素设置样式有了进一步的了解。强调显示当前焦点字段、制作不同背景的下拉菜单等，都是经常用到的，那么本次的上机练习也是针对这些样式的。

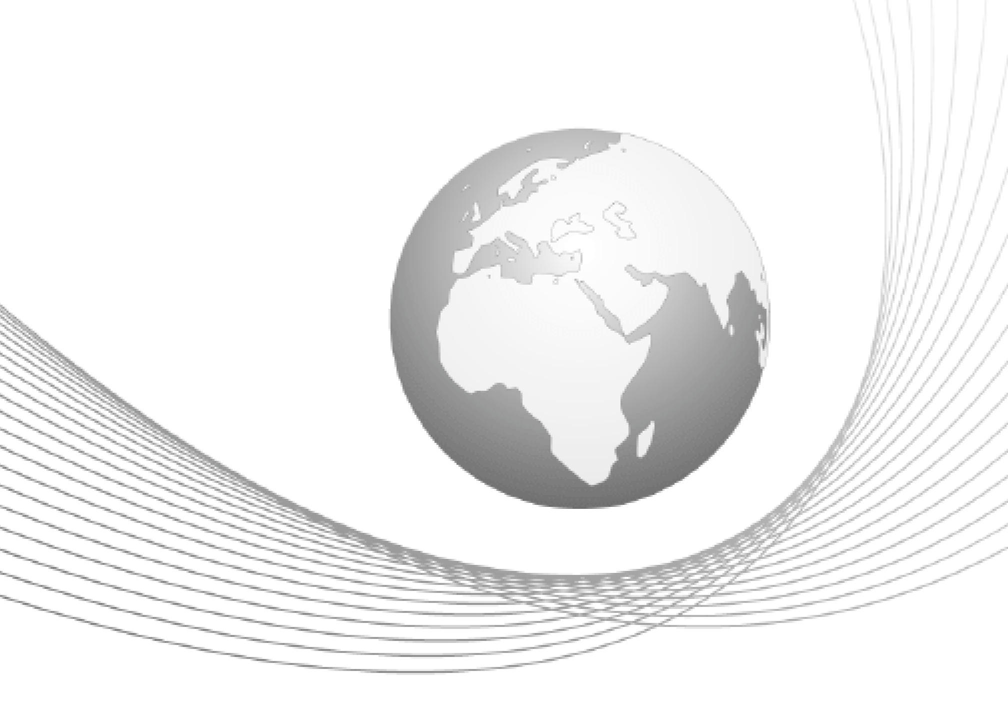
本次练习主要实现，当焦点选中文本框时，边框样式为细边框，边框颜色、背景色均为蓝色，如图 13-17 所示。下拉菜单背景颜色呈“红、橙、黄、绿、青、蓝、紫”等样式，如图 13-18 所示。



图 13-17　强调显示当前的焦点字段



图 13-18　不同背景颜色的下拉菜单



第 14 章　滤镜及其他

内容摘要：

CSS 技术在页面基础样式上具有很好的控制能力，并且具备很好的易用性和扩展性，这使得它在控制页面的同时，还能对页面及其元素进行优化。

随着网页设计技术的发展，人们已经不能仅仅满足于单调地展示页面布局，而是希望在页面中能够加入一些多媒体特效而使页面丰富起来。使用 CSS 滤镜就能够实现这些需求，它能够产生各种各样的文字或图片特效，从而大大地提高了页面的吸引力。

本章将为大家介绍 CSS 中的滤镜特效。

学习目标：

- 掌握 CSS 滤镜特效
- 熟练应用 CSS 滤镜特效
- 掌握如何使用鼠标属性

14.1 滤镜特效

从 Internet Explorer 4.0 开始，浏览器便开始支持多媒体滤镜特效，允许使用简单的代码就能对文本和图片进行处理。比如模糊、彩色投影、火焰效果、图片倒置、色彩渐变、风吹效果和光晕效果等。当把滤镜和渐变结合运用到网页脚本语言中，就可以建立一个动态交互的网页。

滤镜特效在 Web 开发中是必不可少的，不仅结合网页脚本语言能够实现滤镜特效，CSS 在控制页面布局的同时，也提供了多种内置的滤镜特效。使用这种技术可以把可视化的滤镜和转换效果添加到页面元素上。CSS 的滤镜属性的标识符是 filter，它的语法格式如下：

```
filter:filtername(parameters)
```

filtername 是要使用的滤镜名称，如 Alpha、DropShadow 等，如表 14-1 所示列出了几种常用的滤镜名称。parameters 指定了该滤镜中的各参数，通过这些参数才能够决定滤镜显示的效果。

表 14-1　CSS 滤镜

滤镜名称	效　果
Alpha	设置透明度
BlendTrans	实现图像之间的淡入和淡出的效果
Blur	建立模糊效果
Chroma	设置对象中指定的颜色为透明色
DropShadow	建立阴影效果
FlipH	将元素水平翻转
FlipV	将元素垂直翻转
Glow	建立外发光效果
Gray	灰度显示图像，即显示为黑白图像
Invert	图像反相，包括色彩、饱和度和亮度值，类似底片效果
Light	设置光源效果
Mask	建立透明遮罩
RevealTrans	建立切换效果
Shadow	建立另一种阴影效果
Wave	波纹效果
Xray	显现图片的轮廓，类似于 X 光片效果

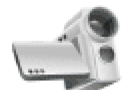


若使用多个滤镜，则每个滤镜之间用空格分隔；一个滤镜中的若干个参数用逗号分隔；filter 属性和其他样式属性并用时以分号分隔。

随着样式表技术的不断成熟，其支持的滤镜种类及功能也逐渐增多。用户如果能够熟练地应用这些滤镜，设计出的网页便会产生意想不到的效果。

14.2 制作文字、图像透明特效

当需要把一张图片放在页面中时，也许会感觉很不“爽”：白色的背景上面放着一张黑色的图片，给人一种“格格不入”的感觉。那么，如何解决这个问题呢？CSS 中的 Alpha 滤镜可以将页面元素与背景混合，从而解决这个问题。下面讲述 Alpha 的滤镜特效。

**视频教学：**光盘/视频/14/ 14.2 制作文字、图像透明特效.avi 长度：5 分钟

14.2.1 基础知识——Alpha 滤镜

Alpha 滤镜是将页面元素与背景混合，从而产生透明渐变的效果。而且可以通过改变数值来控制透明度。如果将 Alpha 滤镜与网页脚本语言结合，并适当地设置其参数，就能使图像显示淡入淡出的效果。

Alpha 滤镜的语法格式如下：

```
{filter : Alpha ( enabled=bEnabled, style=iStyle, opacity=iOpacity,
finishOpacity=iFinishOpacity,
          startx=iPercent, starty=iPercent, finishx=iPercent,
finishy=iPercent ）}
```

各参数如表 14-2 所示。

表 14-2 Alpha 滤镜参数

参 数	说 明
enabled	设置滤镜是否激活
style	设置透明渐变的样式，也就是渐变显示的形状，取值为 0～3。0 表示无渐变，1 表示线形渐变，2 表示圆形渐变，3 表示矩形渐变
opacity	设置透明度，值范围是 0～100。0 表示完全透明，100 表示完全不透明
finishOpacity	设置结束时透明度，值范围也是 0～100
startx	设置透明渐变开始点的水平坐标(即 x 坐标)
starty	设置透明渐变开始点的垂直坐标(即 y 坐标)
finishx	设置透明渐变结束点的水平坐标
finishy	设置透明渐变结束点的垂直坐标

下面为一幅名为 1401 的 GIF 格式的图片设置透明渐变，XHTML 代码如下：

```
<img src="1401.gif" width="150px" height="220px"
style="filter:Alpha(opacity=60,style=0)" />
<img src="1401.gif" width="150px" height="220px"
style="filter:Alpha(opacity=60,style=1)" />
```

```
<img src="1401.gif" width="150px" height="220px"
style="filter:Alpha(opacity=60,style=2)" />
<img src="1401.gif" width="150px" height="220px"
style="filter:Alpha(opacity=60,style=3)" />
```

在代码中设置透明度为 60，并且要求显示各渐变样式，在浏览器中查看滤镜效果，如图 14-1 所示。

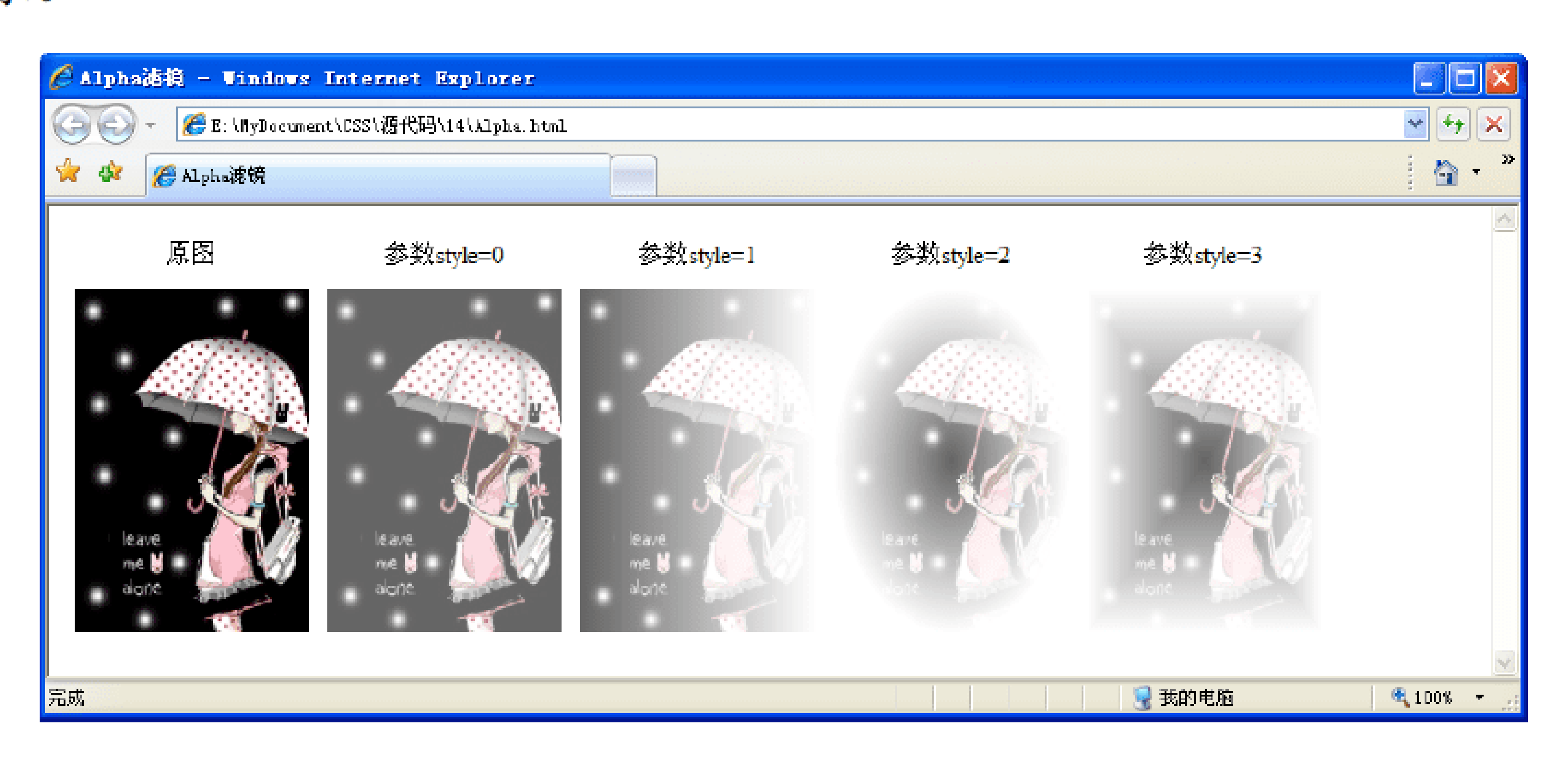


图 14-1　Alpha 滤镜特效

在使用 Alpha 滤镜时要注意以下两点。

- 由于 Alpha 滤镜使当前元素部分透明，使元素下层内容的颜色对整个效果起着重要的作用，因此颜色的合理搭配相当重要。
- 透明度的大小要根据具体情况仔细调整，取一个最佳值。

14.2.2　实例描述

我一个朋友开宠物店的，他雇佣了一个程序员给他做宠物网站。今天，我闲着无事就去他的宠物店找他，他告诉我说：“这个程序员真不行，我让他给我做一个页面，他给我把一张宠物狗的图片往页面上一堆就不管了，真是让我头疼，那页面看得我心酸，真难看”，我打开那个程序员做的网页，确实很难看，于是我做了一个 Alpha 滤镜特效让他看，看是否是他想要的效果，结果还真是。下面就一起来做一个宠物狗滤镜特效页面吧！

14.2.3　实例应用

【例 14-1】制作文字、图像透明特效——宠物展示页面

(1) 新建 HTML 文件，命名为 14-2.html。

(2) 在页面中编辑文字、图像的 Alpha 特效。完整代码如下：

```
<body style="background-image:url(images/2008122717592193.jpg)">
   <div>
      <p style="filter:alpha(opacity=60,style=2)">漂亮宝贝</p>
   </div>
```

```
    <img src="images/9b8e04dd437881e3cc11660b.jpg"  width="200px"
height="240px" />
    <img src="images/9b8e04dd437881e3cc11660b.jpg" width="400px"
height="480px" style="filter:alpha(opacity=60,style=2)" />
</body>
```

上面代码中的背景图片(images/2008122717592193.jpg)是黑色的，宠物狗图片(images/9b8e04dd437881e3cc11660b.jpg)是白色的。为了更全面地让大家认识到 Alpha 滤镜的应用特效，先将“images/9b8e04dd437881e3cc11660b.jpg”图片不加任何修饰地展现在页面中，然后再对其添加 Alpha 滤镜特效。

(3) 保存为 14-2.html 页面。

14.2.4　运行结果

在浏览器中运行 14-2.html 页面，效果如图 14-2 所示。

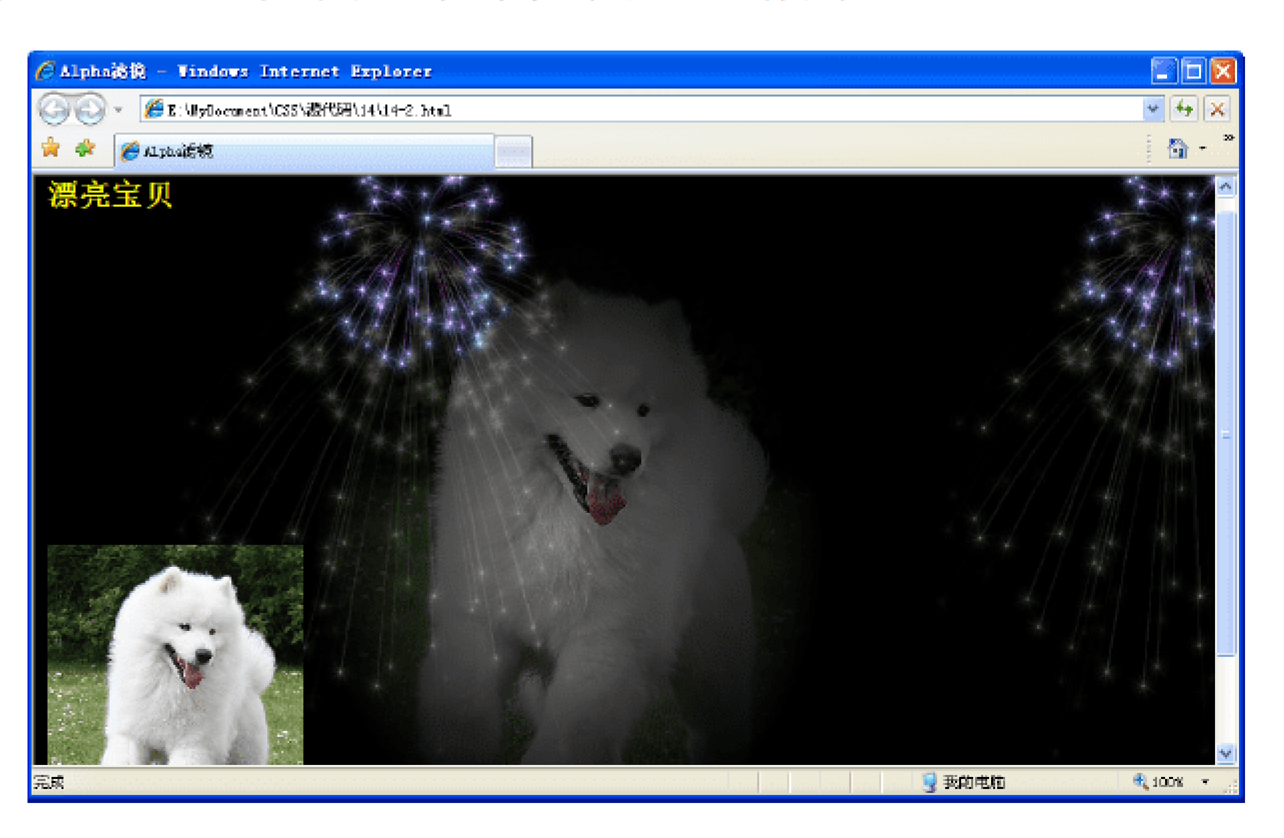


图 14-2　Alpha 滤镜特效

14.2.5　实例分析



源码解析：

在这个实例中，设置了“漂亮宝贝”文字的透明度为 60，圆形渐变样式；并设置了宠物狗图像的透明度也是 60，圆形渐变样式。当然，大家还可以设置页面对象的结束透明度、开始渐变的水平坐标和垂直坐标、结束渐变的水平坐标和垂直坐标等属性，自定义渐变样式。

14.3　制作两张图像轮换交替显示效果

还记得多张图片交替轮换显示吗？当时用的可能是 Flash 吧！如果一个页面上使用了 Flash 播放图片，那么用户在浏览这个网页时，他的电脑上必须安装 Flash 插件，否则网页上的这个

Flash 是无法正常显示的，这个问题也曾经令你苦恼过吧？不用急，CSS 中的 BlendTrans 滤镜特效将会解决这个问题。

**视频教学：**光盘/视频/14/14.3 制作两张图像轮换交替显示效果.avi 长度：4 分钟

14.3.1 基础知识——BlendTrans 滤镜

BlendTrans 滤镜能够实现对象内容的渐隐效果。它是一种淡入淡出的效果滤镜，能产生极精细的图片及文字的转换效果。

BlendTrans 滤镜的语法格式如下：

```
{ filter : BlendTrans ( enabled=bEnabled , duration=fDuration ) }
```

其中，enabled 表示是否激活该滤镜；duration 表示整个转换过程所需要的时间，单位为秒。

要使用 BlendTrans 滤镜实现图片及文字淡入淡出转换的效果，必须借助脚本语言 JavaScript 来调用它的方法。首先，需要使用两幅用来转换的图像，如下面的 JavaScript 代码所示：

```
    imgNum=new ImgArray(2);
    imgNum[0]="images/3.jpg";
    imgNum[1]="images/2.jpg";
```

然后需要编写 JavaScript 的方法来定义转换过程，如下面代码所示：

```
var i=1;
function playImg(){
    if(i==1){i=0;}
    else{i++;}
    imgpic.filters[0].apply();
    imgpic.src=imgNum[i];
    imgpic.filters[0].play();
    timeout=setTimeout('playImg()',4000);
}
```

当然，定义 BlendTrans 滤镜的样式是不可缺少的，定义类样式为：

```
.blendtrans{filter:blendTrans(Duration=3);}
```

在页面中定义上面 JavaScript 脚本中用到的“imgpic”元素，并调用“blendtrans”的类样式，代码如下：

```
<img src="images/3.jpg"  class="blendtrans" id="imgpic"/>
```

这样就完成了图片淡入淡出、仿 Flash 播放多张图片的特效。

14.3.2 实例描述

一个网站的首页就相当于一个人的“脸面”，是非常关键的一个页面，而很多设计者在设计首页内容时，会在页面中展示一个类似于 Flash 播放的效果，但是并非要用到 Flash，因为对

于一个优越的网站，浏览速度也是很关键的，当一个浏览者打开网站需要花费很长时间的时候，可想而知浏览者是不会去看这个网站的。因此，很多设计者都会用 CSS 中的 BlendTrans 滤镜的特效。下面我们就来做一下吧！

14.3.3　实例应用

【例 14-2】制作两张图像轮换交替显示效果

(1) 新建 HTML 文件，命名为 14-3.html。

(2) 两张图片的轮换交替显示功能采用 BlendTrans 滤镜实现，必须借助于脚本语言 JavaScript 来调用它的方法。因此这一步需要在页面中编辑 JavaScript 代码，实现两张图片的轮换交替显示特效，代码如下：

```
<script language="JavaScript">
    //声明数组，数组元素的个数就是图片的个数，然后给数组元素赋值，值为图片路径
    imgNum=new ImgArray(2);
    imgNum[0]="images/3.jpg";
    imgNum[1]="images/2.jpg";
    //获取数组记录数
    function ImgArray(len){
        this.length=len;
    }
    var i=1;
    //转换过程
    function playImg()
    {
        if(i==1){i=0;}
        else{i++;}
        imgpic.filters[0].apply();
        imgpic.src=imgNum[i];
        imgpic.filters[0].play();
        //设置演示时间，这里以毫秒为单位，4000 则表示延迟 4 秒
        //滤镜中设置的转换时间值，这样当转换结束后还停留一段时间
        timeout=setTimeout('playImg()',4000);
    }
</script>
```

(3) 在页面中定义样式，使用 BlendTrans 滤镜特效，CSS 样式代码如下：

```
.blendtrans{
    filter:blendTrans(Duration=3);
}
```

(4) 在页面的<body>标签中定义页面加载事件(onload)，并指定页面加载时，要执行上面定义好的 JavaScript 代码中的 playImg()函数。代码如下所示：

```
<body onload="playImg()">
<img src="images/3.jpg"  class="blendtrans" id="imgpic"/>
</body>
```

14.3.4 运行结果

运行 14-3.html 页面，在浏览器中便可以看到 BlendTrans 滤镜结合 JavaScript 所实现的图片转换效果，分别如图 14-3～图 14-5 所示。

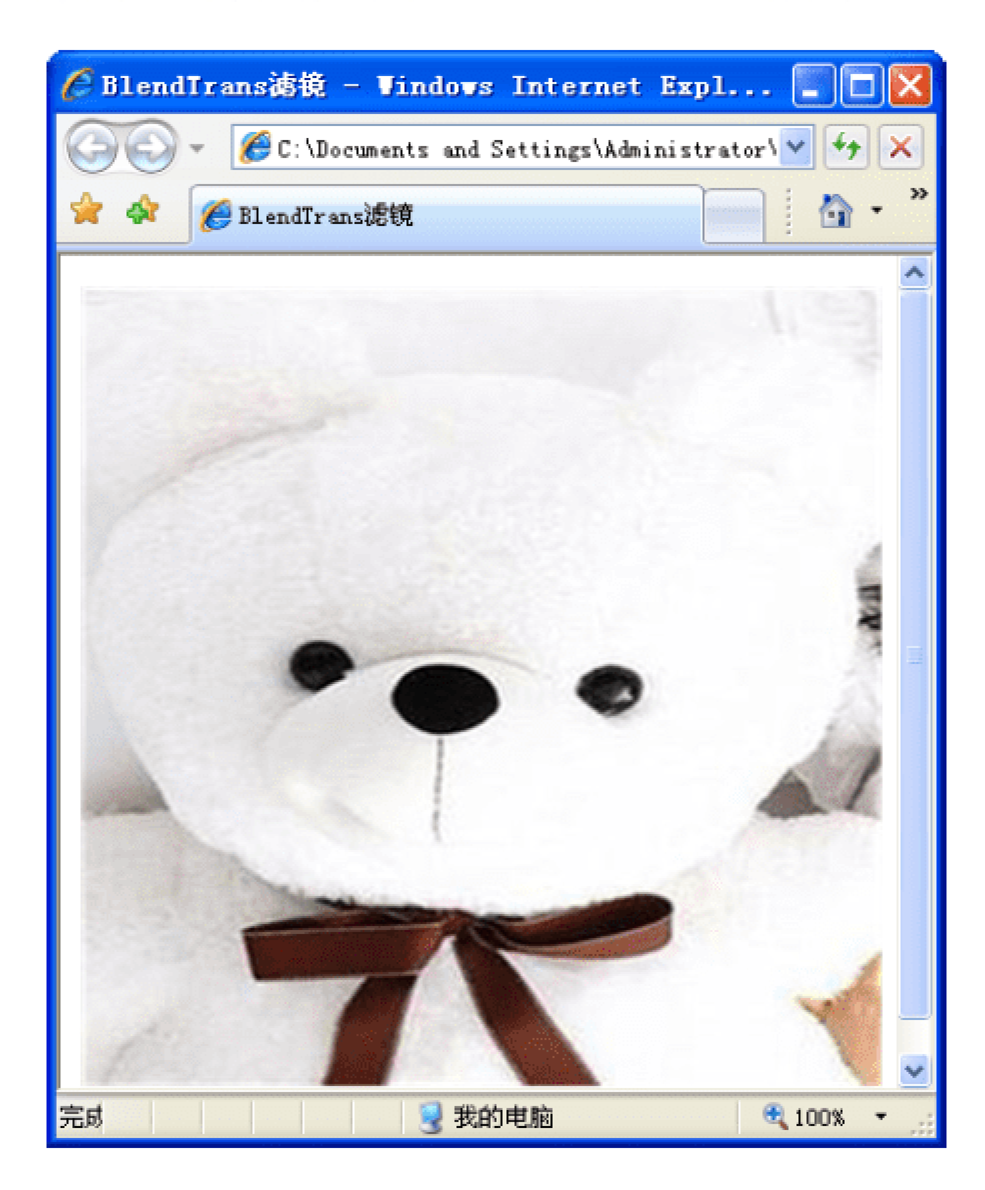


图 14-3 第一幅图片



图 14-4 转换过程中

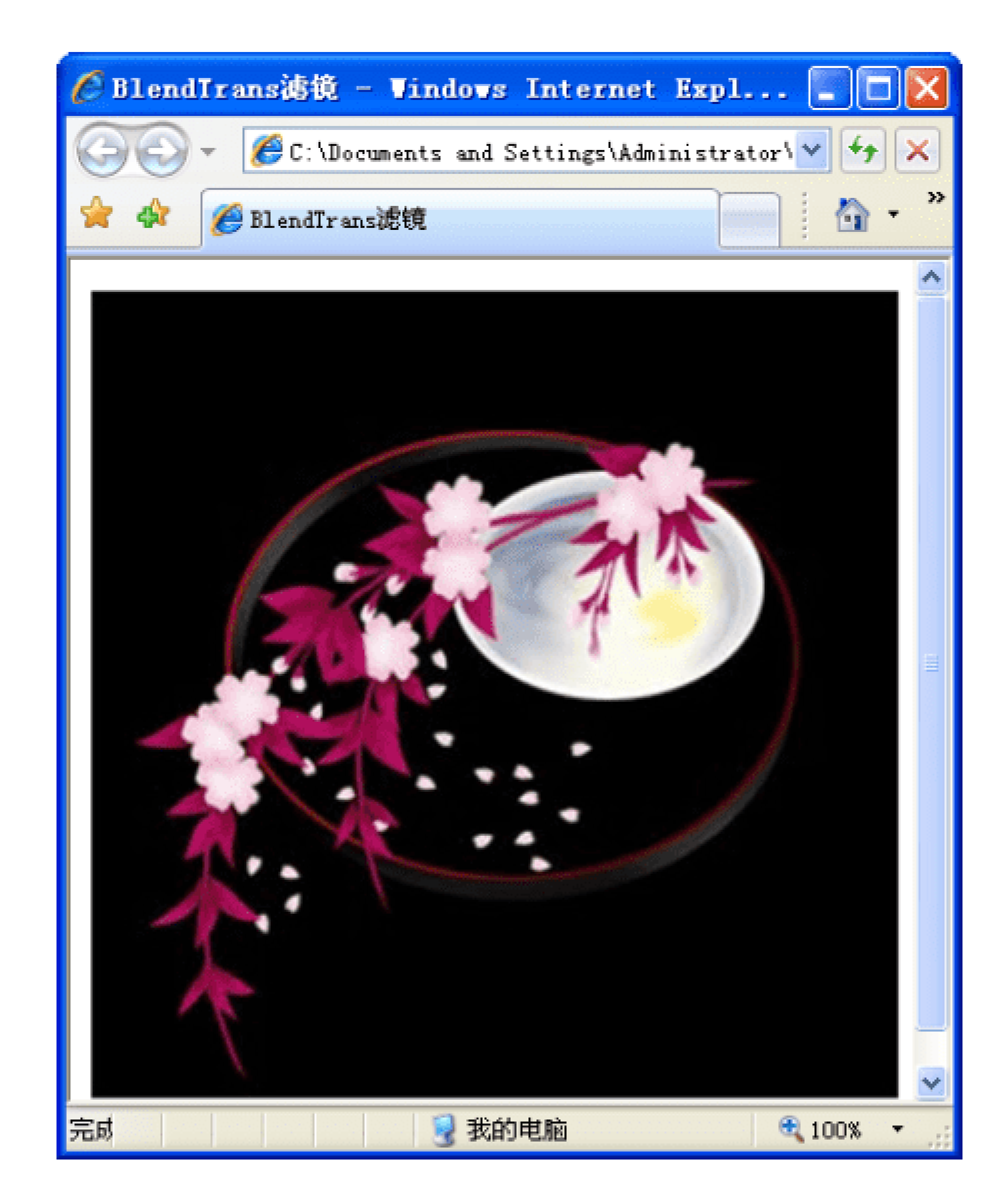


图 14-5 第二幅图片

14.3.5　实例分析



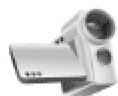
源码解析：

在这个实例中，首先在页面元素 img 中应用 BlendTrans 滤镜，然后使用 JavaScript 脚本语言来定义转换过程。

在 JavaScript 中，首先要声明用来存储图片的数组，并指定图片所在的路径。然后再获取数组的长度，用于转换过程中循环读取图片数量。接着定义转换过程函数——playImg()，该过程实现了两幅图片之间淡入淡出并进行转换的过程，apply 方法用于捕获对象内容的初始显示，为转换做必要的准备，只有调用了 play 方法才开始转换。timeout 指定了转换的延迟时间，再加上滤镜中设置的转换时间，则图片在转换之间将会停留一段时间，以便清楚地浏览图片。最后，在主体元素 body 中插入 onload 事件，加载转换过程。

14.4　制作麦子随风起舞特效

有时候需要使页面元素产生模糊效果，比如一阵风吹过来，需要让麦田里的小麦都跟随风的方向产生运动模糊效果。其实，并非需要大费周折地去做一张“动态”图片，使用 CSS 样式中的 Blur 滤镜就可以让一张静的图片产生“动”的特效。



视频教学：光盘/视频/14/14.4　制作麦子随风起舞特效.avi　　长度：4 分钟

14.4.1　基础知识——Blur 滤镜

Blur 滤镜用于使页面元素产生模糊效果，如果应用得当，就可以产生高速移动的动感效果。Blur 滤镜的语法格式如下：

```
{filter : Blur ( enabled=bEnabled, add=iadd, direction=idirection,
         strength=fstrength )}
```

各参数如表 14-3 所示。

表 14-3　Blur 滤镜参数

参　数	说　明
enabled	设置滤镜是否激活
add	指定图片是否改变成模糊效果。这是个布尔参数，有效值为 True 或 False。True 是默认值，表示应用模糊效果，False 则表示不应用
direction	设定模糊方向。模糊的效果是按顺时针方向起作用的，取值范围为 0～360 度，45 度为一个间隔。有八个方向值：0 表示零度，代表向上方向。45 表示右上，90 表示向右，135 表示右下，180 表示向下，225 表示左下，270 表示向左，315 表示左上

续表

参　数	说　明
strength	指定模糊半径大小，单位是像素，默认值为 5，取值范围为自然数，该取值决定了模糊效果的延伸范围

下面通过一个实例说明 Blur 滤镜的应用及显示效果，代码片段如下：

```
<img src="images/20090527134825718589.jpg"/>
<img src="images/20090527134825718589.jpg"
style="filter:Blur(add=true,direction=225,strength=20)" />
<img src="images/20090527134825718589.jpg"
style="filter:Blur(add=false,direction=225,strength=20)" />
```

在浏览器中查看滤镜效果，如图 14-6 所示。



图 14-6　Blur 滤镜效果

14.4.2　实例描述

我们来想象一种场景，一阵风吹过后，麦田里的麦子随风起舞，这是一种多么美丽的景象啊。当然在网站开发的过程中也常常需要这样的场景来修饰页面，下面就来制作麦子随风起舞的特效吧！

14.4.3　实例应用

【例 14-3】制作麦子随风起舞特效

(1) 新建 HTML 文件，命名为 14-4.html。

(2) 在页面中定义第一个层，在层中插入一张图片，并使用 Blur 滤镜特效，使这张图片实现“风吹”的效果。代码如下：

```
<div id="apDiv1">
    <img src="images/3dd8b236bfc707eaa2cc2b15.jpg"
```

```
        style="filter:blur(add=true,direction=225,strength=30)"/>
</div>
```

(3) 在页面中定义第二个层，在这个层中写入“风好大啊！”字体，并设置该层引用的样式，让字体实现 Blur 特效。代码如下：

```
<style type="text/css">
#apDiv2 {
    position:absolute;
    width:253px;
    height:115px;
    z-index:2;
    left: 643px;
    top: 5px;
    font-size:40px;
    color:#006633;
    font-weight:bold;
    font-family:"华文行楷";
    filter:blur(add=false,direction=90,strength=10);
}
</style>
<div id="apDiv2">风好大啊！</div>
```

14.4.4　运行结果

在浏览器中查看 14-4.html 页面的运行效果，如图 14-7 所示。



图 14-7　麦子风吹效果

14.4.5 实例分析



源码解析：

通过这个实例，了解到 Blur 滤镜不仅可以应用于图片，还可以应用于文字，使文字产生类似于风吹的效果，如图 14-7 所示。挺拔的麦子经过“Blur”滤镜应用后，显示出风吹麦子的动态效果。

14.5 制作半透明字体

有时候需要对页面中的文字进行修饰，使其变得美观大方、同时又极富有个性，这时，可以选择将其文字中指定的颜色设置为透明色。CSS 中的 Chroma 滤镜就可以实现这样的效果。



视频教学：光盘/视频/14/14.5 制作半透明字体.avi　　**长度：**4 分钟

14.5.1 基础知识——Chroma 滤镜

Chroma 滤镜可以将对象中指定的颜色设置为透明色。在页面元素中将该滤镜应用于对象时，则对象内容中的指定颜色就变为透明的，即不可见了。其语法格式如下：

```
{filter : Chroma(enabled=bEnabled , color=sColor)}
```

其中，color 参数设置要变为透明色的颜色。

下面通过一个小实例来演示 Chroma 滤镜的应用，在页面中编辑如下代码：

```
<style>
      div{position:absolute;top:70;letf:40; filter:Chroma(color=red)}
      p{font-size:30pt; font-weight:bold; color:red}
</style>
<p>Chroma 滤镜效果</p>
   <div>
      <p>Chroma 滤镜效果</p>
   </div>
```

代码中定义了元素 div 的绝对位置，并设置该元素的范围内红色为透明色。在段落元素 p 中的文字显示为红色，在应用了滤镜后，效果如图 14-8 所示。拖动鼠标选择过滤颜色后的文字，便可以查看过滤掉颜色的文字，如图 14-9 所示。

Chroma 滤镜一般应用于文字特效，而且对于有些格式的图片也是不适用的。例如，JPEG 格式的图片是一种已经减色和压缩处理的图片，所以要设置其中某种颜色透明十分困难。此外，每幅图片只能指定一种透明色。对于已设置另外的色为透明色的 GIF 等格式的图片，再设置透明色时，原先透明而不可见的颜色就会重新显示出来。



图 14-8　Chroma 滤镜效果



图 14-9　显示过滤颜色的文字

14.5.2　实例描述

我是一个很喜欢宠物的人，所以就希望每张宠物图片都做到“完美无瑕”。今天，心血来潮，就做了一个“漂亮宝贝”的宠物秀。下面就来给大家分享一下吧！

14.5.3　实例应用

【例 14-4】制作宠物秀页面

(1) 新建 HTML 文件，命名为 14-5.html。

(2) 编辑 14-5.html 页面，在页面中定义一个层展示宠物，并在层上面写 “漂亮宝贝”四个字，字体要求是透明色的，黑边框。完整代码如下：

```
<body>
    <div style=" width:850px; height:680px;position:absolute;top:70;letf:40;
filter:Chroma(color=black);
background-image:url(images/20061102171207491859.jpg);
background-repeat:no-repeat;">
        <p style="font-size:28pt; font-weight:bold; color:black;
position:relative; left:100px; top:100px;">漂亮宝贝</p>
    </div>
</body>
```

(3) 保存 14-5.html 页面。

14.5.4　运行结果

在浏览器中查看“宠物秀”页面，效果如图 14-10 所示。

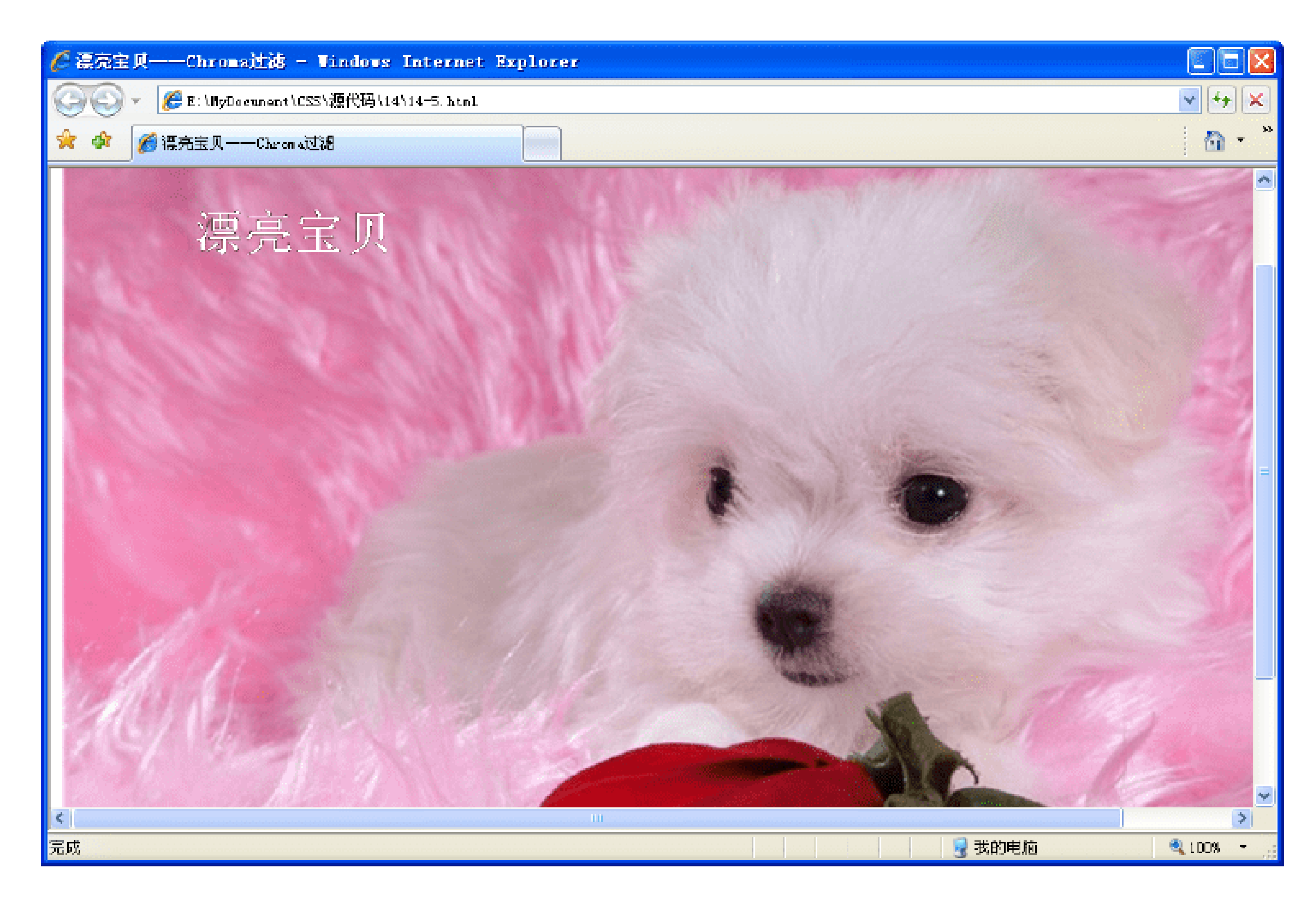


图 14-10 宠物秀

14.5.5 实例分析

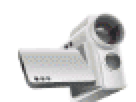


源码解析：

在该实例中，定义了 p 标签中的“漂亮宝贝”字体为黑色、加粗、28 像素，而在嵌套在它外面的 div 标签中定义了 Chroma 滤镜，并设置了要变为透明色的颜色是黑色，因此会出现黑边框、透明色的字体，如图 14-10 所示的那样。

14.6 制作投影“宠物秀”页面

很多时候需要使页面元素内容在页面上产生投影，从而实现立体的效果。CSS 中的 Shadow 滤镜和 DropShadow 滤镜就可以帮你实现。



视频教学：光盘/视频/14/14.6 制作投影“宠物秀”页面.avi 长度：6 分钟

14.6.1 基础知识——Shadow 滤镜与 DropShadow 滤镜

Shadow 滤镜和 DropShadow 滤镜都可以为页面对象设置投影。唯一不同的就是 Shadow 滤镜能够在指定的方向设置投影，且投影可产生渐进的效果，而 DropShadow 滤镜是利用偏移量来定义投影位置。下面就来详细地介绍一下这两个滤镜。

1. Shadow 滤镜

Shadow 滤镜可以在指定的方向建立对象的投影，且投影可产生渐进的效果，再结合实体

更有立体的感觉。它的语法格式如下：

```
{filter:Shadow(enabled=bEnabled, color=sColor, direction=iOffset, strength=iDistance)}
```

各参数如表 14-4 所示。

表 14-4　Shadow 滤镜参数

参　数	说　明
enabled	设置滤镜是否激活
Color	设置投影的颜色
direction	设置投影的方向，分别有 8 种取值代表 8 种方向：取值为 0 表示向上方向，45 为右上方，90 为右方，135 为右下方，180 为下方，225 为左下方，270 为左方，315 为左上方
strength	设置投影向外扩散的距离

2. DropShadow 滤镜

DropShadow 滤镜用于建立阴影效果。该滤镜使元素内容在页面上产生投影，从而实现立体的效果。其工作原理就是创建一个偏移量，并定义一个阴影颜色，使之产生效果。

DropShadow 滤镜的语法格式如下：

```
{filter : DropShadow ( enabled=bEnabled , color=sColor , offx=iOffsetx,
offy=iOffsety, positive=bPositive )}
```

语法中各参数如表 14-5 所示。

表 14-5　DropShadow 滤镜参数

参　数	说　明
enabled	设置滤镜是否激活
color	指定滤镜产生的阴影颜色
offx	指定阴影水平方向偏移量，默认值为 5px
offy	指定阴影垂直方向偏移量，默认值为 5px
positive	指定阴影透明程度，为布尔值。True(1)表示为任何的非透明像素建立可见的阴影；False(0)表示为透明的像素部分建立透明效果

14.6.2　实例描述

制作宠物秀页面时，可以在一张很漂亮的宠物图片上面编辑透明字体来优化宠物图片，还可以使用投影效果更进一步地优化宠物秀图片，不仅可以使宠物图片实现投影效果，还可以在宠物图片上面编辑字体，并实现投影效果。下面就来做一做吧！

14.6.3　实例应用

【例 14-5】制作投影“宠物秀”页面

(1) 新建 HTML 文件，命名为 14-6.html。

(2) 编辑 14-6.html 页面，在页面中定义一行两列的表格，在两个单元格中分别插入一张宠物图片和一个层，其中一个单元格的层中写着“可爱宝贝”的字样，另一个单元格的层中写着“漂亮宝贝”的字样。表格代码如下：

```
<table cellpadding="5" cellspacing="0" border="0" align="center">
    <tr>
        <td class="shadow">
            <img src="images/20061102171207491859_1.jpg"/>
            <div id="Div1"> 可爱宝贝</div>
        </td>
        <td class="dropshadow">
            <img src="images/79e0fbcbdadbf85c7f3e6fb2.jpg" />
            <div id="Div2">漂亮宝贝</div>
        </td>
    </tr>
</table>
```

(3) 在上面的表格中总共有四项内容，两张宠物图片和两个层，下面编辑这四项内容的样式，使它们实现 Shadow 滤镜特效。CSS 样式如下：

```
td{
    text-align:center;
}
.shadow {
    padding-bottom:10px;
    font-weight:bold;
    font-size:24px;
    filter: shadow(color = gray, direction =135,strength=20);
}
.dropshadow {
    font-weight:bold;
    font-size:24px;
    filter: dropshadow(color = gray, offx = 15, offy = 15, positive = 1);
}
```

14.6.4　运行结果

在浏览器中运行 14-6.html 页面，出现如图 14-11 所示的页面。

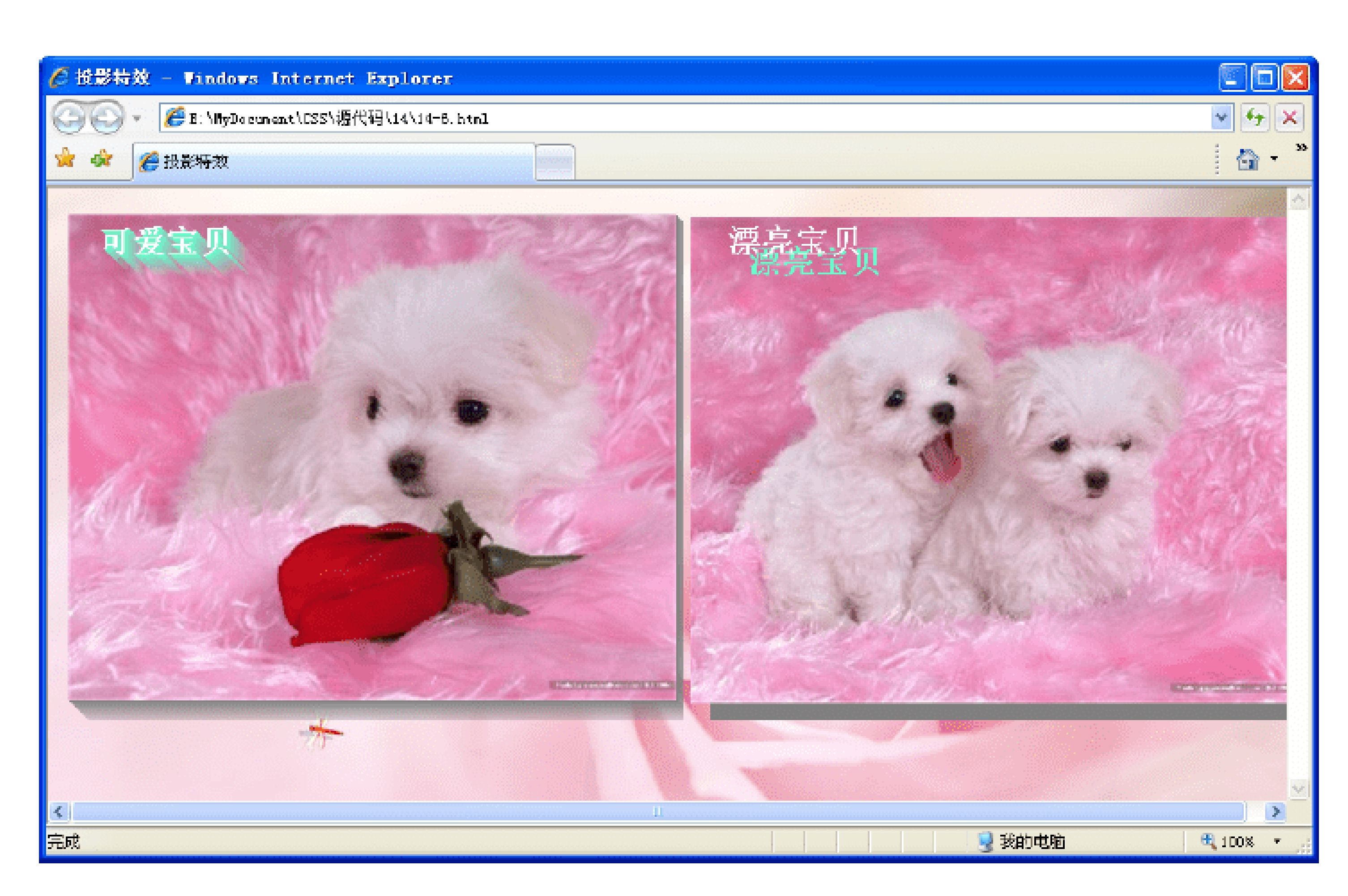


图 14-11　Shadow 滤镜和 DropShadow 滤镜的效果

14.6.5　实例分析

**源码解析：**

在这个实例中，定义了“shadow”和“dropshadow”两个类选择符，在页面中定义了一个一行两列的表格，表格的两个单元格分别应用了“shadow”选择符和“dropshadow”选择符，因此表格中的图像有了投影的效果。接着又在两个单元格中分别定义了层布局，并定义了“Shadow”投影效果和“DropShadow”投影效果，故层中的文字实现了投影的效果。

14.7　可翻转的图像和文字

当需要把一张图片水平翻转 180°，也就是希望看到的是这张图片的“背面”时，是否需要大动干戈地去做该张图片的“背面图像”呢？当需要把一张图片垂直翻转 180°，也就是希望浏览者“站在电脑顶部往下看”，那是否也需要大费周折地去做一张图片呢？不需要，CSS 为大家提供了 FlipH 滤镜和 FlipV 滤镜，这两个滤镜就可以完成这样的工作。

好奇者或许也很想知道“祖国”的倒体字是怎样的吧？这些效果使用 CSS 中的 FlipH 滤镜和 FlipV 滤镜都可以完成。

**视频教学：**光盘/视频/14/14.7　可翻转的图像和文字.avi　　　长度：5 分钟

14.7.1　基础知识——FlipH 滤镜与 FlipV 滤镜

需要使页面元素水平翻转或者垂直翻转，不需要做一张翻转后的图片放在页面中；如果想知道某些文字的倒体字也不难，CSS 中的 FlipH 滤镜和 FlipV 滤镜都可以完成这些复杂的工作。

下面就具体地讲述这两个滤镜的用法。

1. FlipH 滤镜

FlipH 滤镜用于水平翻转对象，即将元素对象按水平方向进行 180 度翻转。FlipH 滤镜可以在 CSS 中直接使用，使用格式如下：

```
{Fliter: FlipH(enabled=bEnabled)}
```

该滤镜中只有一个 enabled 参数，表示是否激活该滤镜。

FlipH 滤镜的使用非常简单，只需要在页面元素中添加 filter:FlipH()即可，比如需要使 1402.jpg 的图片水平翻转，代码如下：

```
<img src="1402.jpg" style="filter:FlipH()" />
```

在浏览器中查看效果，如图 14-12 所示。

FlipH 滤镜除了能够实现图片的翻转外，还能够实现文字的翻转，例如，下面代码实现了文字翻转的效果。运行结果如图 14-13 所示。

```
<p style="filter:FlipH()">祖国</p>
```



图 14-12　FlipH 滤镜效果(图像水平翻转)

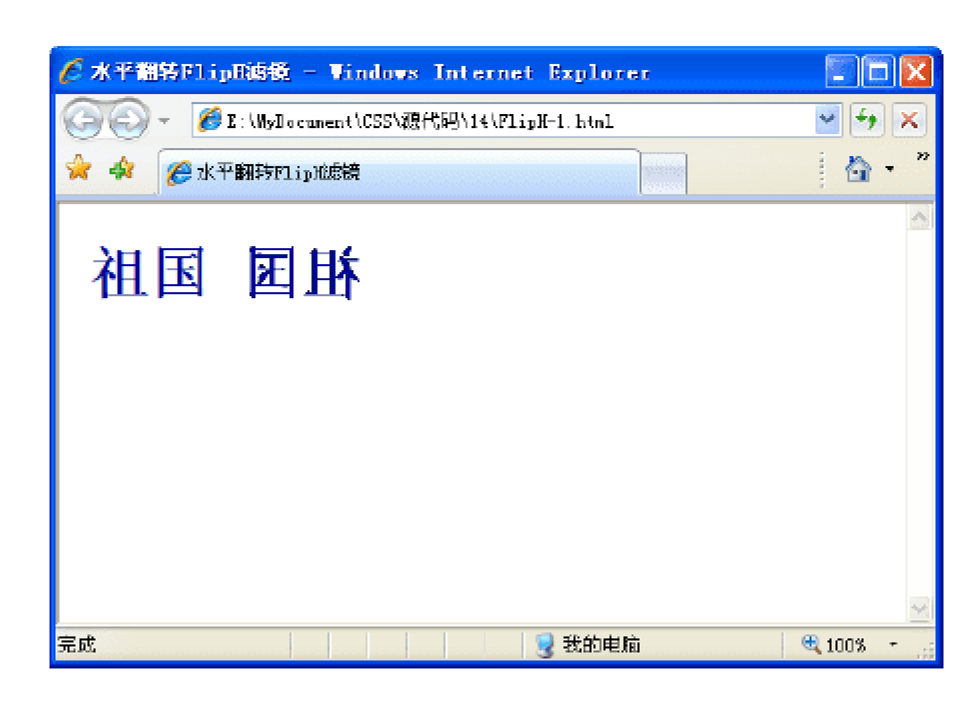


图 14-13　文字水平翻转

2. FlipV 滤镜

FlipH 滤镜用来实现对象的水平翻转，而 FlipV 滤镜用来实现对象的垂直翻转，其中包括文字和图像。语法格式如下：

```
{Fliter: FlipV(enabled=bEnabled)}
```

该滤镜只包含有一个表示是否激活滤镜的参数，能够在代码中直接使用。例如使 1402.jpg 图像垂直翻转，可以使用下面的代码：

```
<img src="1402.jpg" style="filter:FlipV()" />
```

在浏览器中查看效果，如图 14-14 所示。

若要实现文字的垂直翻转，则可以使用下列代码：

```
<p style="filter:FlipV()" >祖国</p>
```

在浏览器中查看滤镜效果，如图 14-15 所示。



图 14-14　FlipV 滤镜效果(图片垂直翻转)

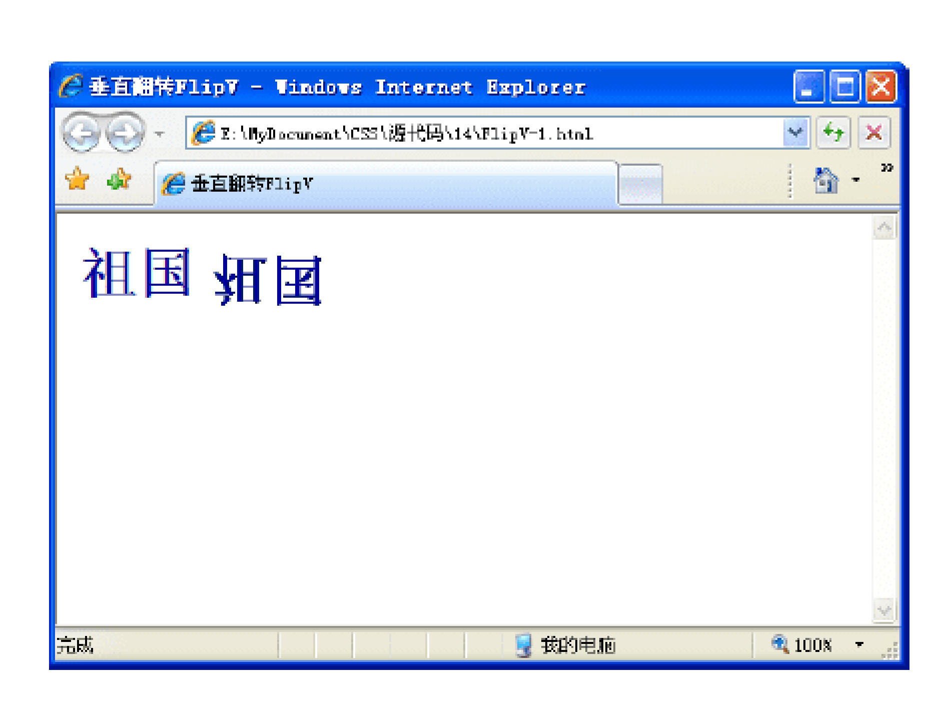


图 14-15　文字垂直翻转

14.7.2　实例描述

可以想象一下，如果有两个人相对弹琴，那将是多么美妙的景象啊！而现在只有一张面朝左边的弹琴图片，那么下面我们就来做一张面朝右边的图片吧！

14.7.3　实例应用

【例 14-6】两人相对弹琴

(1) 新建 HTML 文件，命名为 14-7.html。

(2) 编辑 14-7.html 页面，在页面中添加一个 id 为“Div2”的层，在该层中使用<img>标签引入一张正在弹琴的古人图片，代码如下：

```
<div id="Div2">
    <img src="images/017046ef2f0cb3e4b3fb950d.png" width="500" height="388" />
</div>
```

(3) 定义“Div2”层的样式，CSS 代码如下：

```
#Div2 {
        position:absolute;
        width:500px;
        height:388px;
        z-index:2;
        left: 525px;
        top: 229px;
    }
```

(4) 在 14-7.html 页面中再添加一个 id 为“Div3”的层，在该层中也引入一张“Div2”层中引入的图片，并在<img>标签中定义“style”样式，使该图像实现 FlipH 滤镜特效，代码如下：

```
<div id="Div3">
    <img src="images/017046ef2f0cb3e4b3fb950d.png" width="500" height="388"
style="filter:FlipH()"/>
</div>
```

(5) 定义“Div3”层的样式，CSS 代码如下：

```
#Div3 {
        position:absolute;
        width:500px;
        height:388px;
        z-index:1;
        left: 0px;
        top: 229px;
    }
```

(6) 设置页面背景色为黑色，即在<body>标签中添加样式，代码如下所示：

```
<body style="background-color:black;">
    ...
</body>
```

14.7.4 运行结果

在浏览器中查看 14-7.html 页面效果，如图 14-16 所示。



图 14-16 两人相对弹琴(图像水平翻转)

14.7.5 实例分析



源码解析：

在这个实例中，定义了两个层来放置“017046ef2f0cb3e4b3fb950d.png”图片，一张是原图，也就是“Div2”层中的图像(图 14-16 中右边的图像)；另一张是经过翻转之后的图像，即“Div3”层中的图像(图 14-16 中左边的图像)，从而形成了两个古代美女弹琴的景象。

14.8　制作光晕字

大家都喜欢很柔和的感觉，给人一种舒适、安详的美感。CSS 在这点上一点也不逊色，它提供了 Glow 滤镜，可以使对象的边缘产生一种柔和的边框或光晕。

 **视频教学：**光盘/视频/14/14.8　制作光晕字.avi　　　　**长度：**5 分钟

14.8.1　基础知识——Glow 滤镜

Glow 滤镜可以使对象的边缘产生一种柔和的边框或光晕，并可以产生如火焰一样的效果。这种效果的颜色和强度可通过数值进行定义，其语法格式如下：

```
{filter : Glow ( enabled=bEnabled , color=sColor , strength=iDistance ) }
```

其中，color 设置边缘光晕的颜色，strength 设置晕圈的范围，值的范围是 1～255，值越大效果越强。

当 Glow 滤镜作用于文字时，每个文字的边缘都会出现光晕，效果非常强烈。而对于图片，Glow 滤镜只在其边缘上加上光晕效果。下面就以一个实例来演示 Glow 滤镜的特效。

```
    <style>
    <!--
      .leaf{
          width:100%;
          padding-top:10px;
          padding-bottom:10px;
          padding-left: 20px;
          filter:Glow(color=#FF3399,strength=15)}
     .weny{
          width:100%;
          padding-top:10px;
          padding-left:20px;
          filter:Glow(color=#9966CC,strength=10)}
    -->
    </style>
<body style="background-image:url(images/20081214018479377l.jpg);
background-repeat:no-repeat;">
    <div class="leaf">
          <img src="images/header_bg2.jpg" />
    </div>
    <div class="weny">
       <p style="font-family: lucida handwriting; font-size: 32pt; font-weight:
bold; color: #003366">
```

```
        Welcome To My Hometown! </p>
    </div>
</body>
```

代码中，第一个 div 标记使用类选择符 leaf，为图片添加光晕。第二个 div 标记使用类选择符 weny，为文字添加光晕。并且为了使光晕显示完全，都定义了间隙及宽度。在浏览器中查看滤镜效果，如图 14-17 所示。

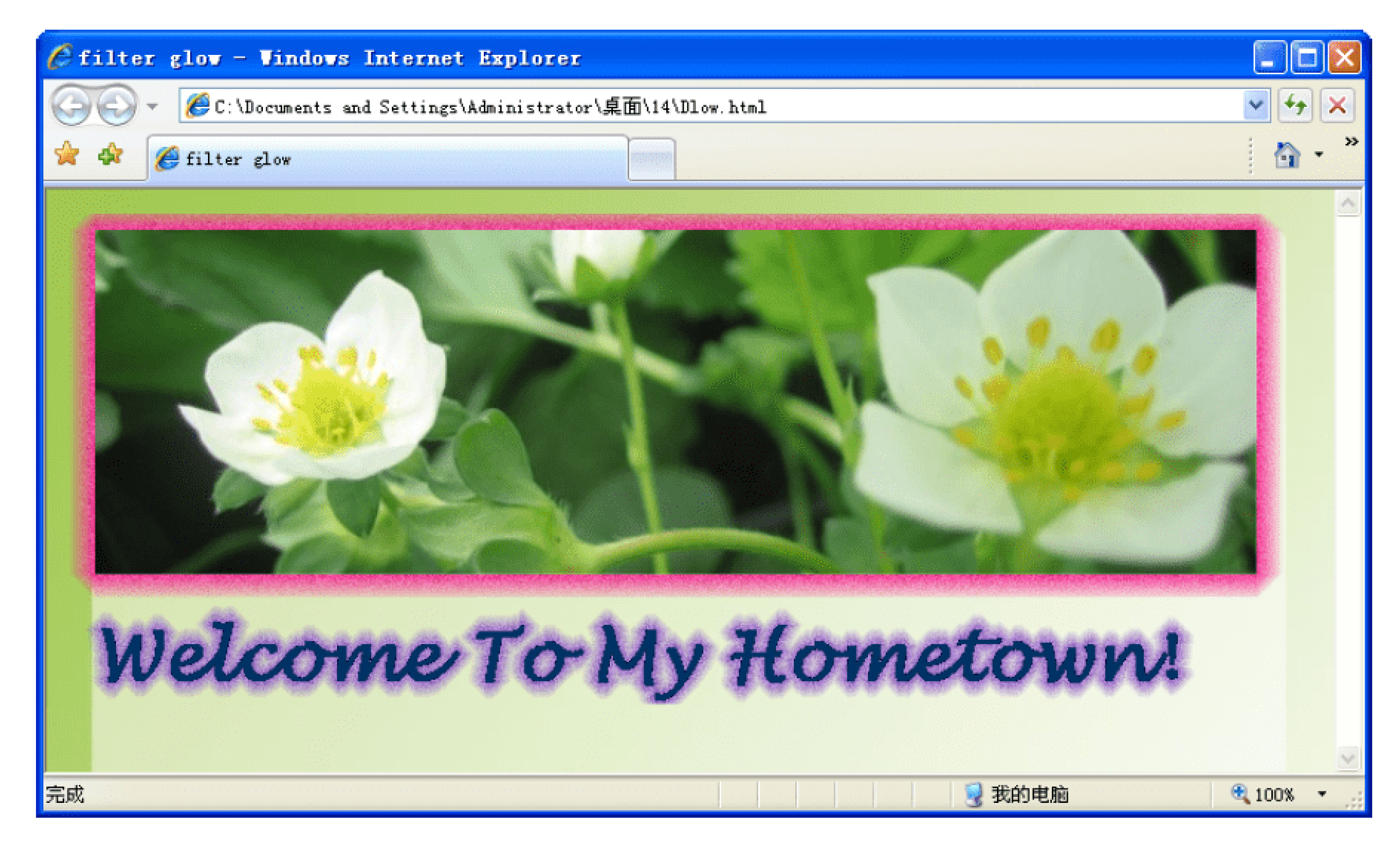


图 14-17　Glow 滤镜效果

14.8.2　实例描述

人世间最伟大的爱莫过于父母之爱，在感恩节来临之际，我们来做一个“感恩父母”的宣传页吧。为了使宣传页显得更加漂亮，这里使用 Glow 滤镜特效给标题边缘添加光晕的特效。下面就来看看具体的实现步骤吧。

14.8.3　实例应用

【例 14-7】制作“感恩父母”宣传页

(1) 新建 HTML 文件，命名为 14-8.html。

(2) 在页面中定义“感恩父母”宣传页的标题文字，代码如下：

```
<p class="china" align="center" style="font-family: 华文行楷; font-size:50pt;
font-weight: bold; color: black;"><br>
感恩父母！！</p>
```

(3) 在页面中定义宣传页主题图片，即在层中引入一张图片，代码如下：

```
    <hr style="border:#000000 solid double"/>
    <div class="chinaPic">
        <img src="images/46d7709356e3622ce333af4e359a9b47.jpg" width="200"
height="150"/>
    </div><br />
```

(4) 在页面中定义宣传页内容，代码如下：

```
<div id="Div1" align="left">
   ……<br />
   感恩的心<br />
   感谢有你<br />
    我来自偶然<br />
    像一颗尘土<br />
    有谁看出我的脆弱<br />
    我来自何方<br />
    我情归何处<br />
    谁在下一刻呼唤我<br />
    ……
</div>
```

(5) 上面定义的这些内容在页面中都是居中显示的，即在这些内容前面和后面添加<center>标签和</center>标签。

(6) 让宣传页的主题和主题图片实现 Glow 滤镜特效，代码如下：

```
    .china{
        filter:Glow(color = #FF3399, strength = 15);
        width: 100%;
        height: 161px;
        top: 23px;
    }
    .chinaPic{
        width:100%;
        padding-top:10px;
        padding-bottom:10px;
        padding-left: 10px;
        filter:Glow(color=yellow,strength=15)
}
```

14.8.4　运行结果

在浏览器中运行 14-8.html 页面，“感恩父母”的宣传页就呈现出来了，如图 14-18 所示。

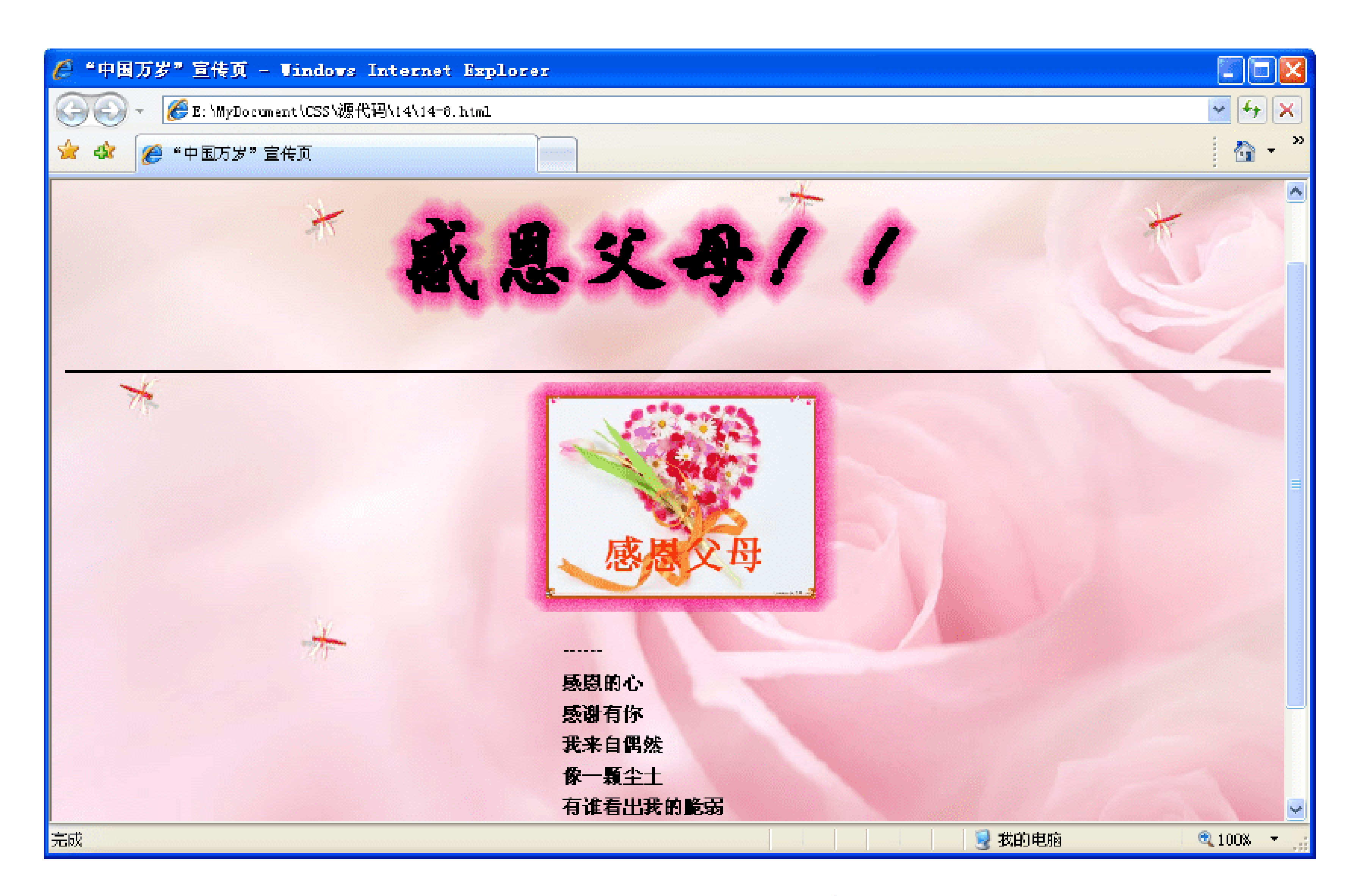


图 14-18　“感恩父母”宣传页

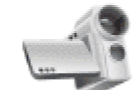
14.8.5　实例分析

**源码解析：**

在这个实例中，指定“感恩父母”四个字使用 Glow 滤镜特效：边缘光晕的颜色为“#FF3399”，晕圈的范围为 15；并指定感恩父母的标识图片也使用 Glow 滤镜特效。这里需要说明一点，如果需要使图片应用光晕特效，在<img>标签中添加样式是不可以的，只能在<img>标签所在的容器<div>中添加样式，如上面实例中的国旗图片容器调用“chinaPic”选择符，而不是在标签<img>中调用。

14.9　让古代美女的粉色衣裳变为灰色

就像是一张色彩鲜明的战争图像转变为灰度图像才能达到复古的感觉一样，灰度图像特效在 Web 开发中也是必不可少的，而 CSS 中的 Gray 滤镜就可以让图像中的色彩全部由黑、白、灰三种颜色代替。下面我们就来详细地介绍 Gray 滤镜的应用。

**视频教学：**光盘/视频/14/14.9　让古代美女的粉色衣裳变为灰色.avi　　长度：8 分钟

14.9.1　基础知识——Gray 滤镜

Gray 滤镜一般应用于图像，用来将一幅图像转变为灰度图，即图像中的色彩全部由黑、白、灰三种颜色代替。

Gray 滤镜的语法格式比较简单，只有一个表示是否激活的参数 enabled。可以在页面代码中直接使用，格式如下：

```
{filter:Gray(enabled=bEnabled)}
```

Gray 滤镜的应用只需要给图像添加样式“filter:Gray()”就可以了，如下面的代码：

```
<img src="images/1120080093.jpg" width="240" height="210"
style="filter:Gray()"/>
```

在浏览器中查看滤镜效果，如图 14-19 所示。



图 14-19　Gray 滤镜效果

14.9.2　实例描述

如今回想古代的事物，都是灰色的，因为没有经历过那种年代，不可能真实地回想起一些什么。那么，现在假设一位古代美女穿着粉色的衣服，含情脉脉地看着远处的风景，这时可以把脑海中想象到的事物编辑成灰色图像展现在页面中，这样会达到更理想的效果。

14.9.3　实例应用

【例 14-8】让古代美女的粉色衣裳变为灰色

(1) 新建 HTML 文件，命名为 14-9.html。

(2) 编辑 14-9.html 页面，页面内容很简单，只是一张简单的图片，并应用 Gray 过滤特效，关键代码如下：

```
<body>
<img src="images/714ebe1561783c35c83d6d29.jpg" style="filter:Gray()" />
</body>
```

(3) 保存页面，查看效果。

14.9.4 运行结果

在浏览器中运行 14-9.html 页面，如图 14-20 所示。



图 14-20 古代美女粉色衣裳变为灰色效果图

14.9.5 实例分析


源码解析：

在该实例中，只是在<img>标签中添加“style="filter:Gray()"”代码，从而使古代美女图像应用 Gray 滤镜特效。原图的美女衣裳是粉色的，但其应用 Gray 滤镜之后变为灰色，由此可见，Gray 滤镜的使用方法非常简单，不仅可以在<img>标签中这样使用，还可以采用在 div、td 等容器中添加“style="filter:Gray()"”代码的方式使用，而无须在<img>标签中添加 Gray 滤镜特效。

14.10 图像大翻身

有时候需要把图像或者文字的可视化属性翻转一下，从而达到自己想要的结果；而有时候需要让页面中的元素轮廓加亮显示，产生类似于 X 光片的效果。这些效果使用 CSS 中的 Invert 滤镜和 Xray 滤镜就可以实现。下面就介绍 CSS 中的 Invert 滤镜和 Xray 滤镜的应用技术。


视频教学：光盘/视频/14/ 14.10 图像大翻身.avi 长度：5 分钟

14.10.1　基础知识——Invert 滤镜与 Xray 滤镜

如果需要制作一张类似于底片效果的图像，使用 CSS 中的 Invert 滤镜就可以做到，该滤镜可以把对象的可视化属性全部翻转。与此相似的还有一个 Xray 滤镜(X 射线)，该滤镜只是让对象反映出它的轮廓，并将这些轮廓加亮，类似于 X 光片效果。下面就来详细介绍如何使用这两个滤镜。

1. Invert 滤镜

Invert 滤镜可以把对象的可视化属性全部翻转，包括色彩、饱和度和亮度值。该滤镜一般用于处理图像，使图像显示类似于底片的效果。

Invert 滤镜的语法格式中只包含一个 enabled 参数，用来设置是否激活滤镜，格式如下：

```
{filter:Invert(enabled=bEnabled)}
```

在需要使用 Invert 滤镜特效的页面对象中添加样式“filter:Invert()”，该对象就实现了 Invert 滤镜特效，如下面的代码所示：

```
<img src="images/5243525_24.jpg" style="filter:Invert()" />
```

在浏览器中查看滤镜效果，如图 14-21 所示。

2. Xray 滤镜

Xray 的中文含义为 X 射线，Xray 滤镜就是让对象反映出它的轮廓并将这些轮廓加亮，产生的效果类似于 X 光片。

Xray 滤镜只有一个 enabled 参数，用于确定是否激活该滤镜，格式如下：

```
    {filter:Xray(enabled=bEnabled)}
```

Xray 滤镜一般应用在图像特效上，实例代码如下所示：

```
<img src="images/5243525_24.jpg" style="filter:Xray()" />
```

在浏览器中查看滤镜效果，如图 14-22 所示。

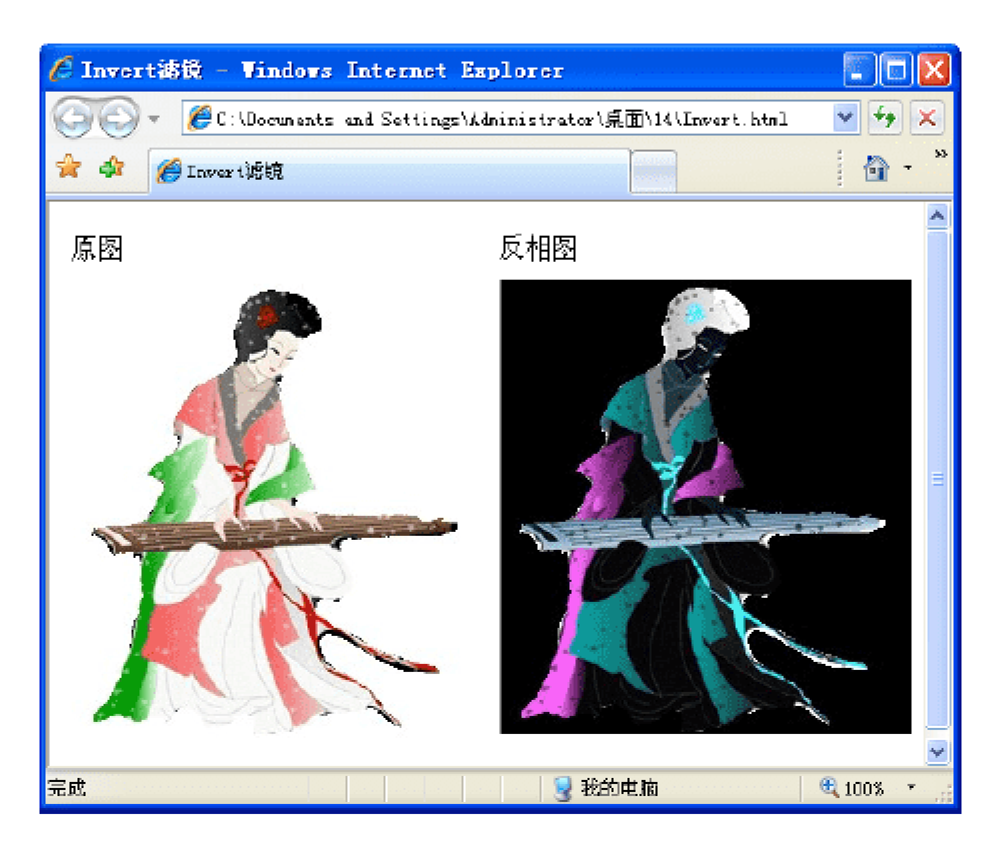


图 14-21　Invert 滤镜效果



图 14-22　Xray 滤镜效果

14.10.2 实例描述

当翻看使用胶卷式的相机拍的照片时，我们只能看到一个人的轮廓。这种效果与使用 Xray 滤镜特效的效果有点相似，下面我们就使用 Xray 滤镜特效来模拟制作相册底片效果吧！

14.10.3 实例应用

【例 14-9】制作相册底片效果

(1) 新建 HTML 文件，命名为 14-10.html。

(2) 在页面中定义一个三行一列的表格，第一行为一张类似于胶卷底片边框的图片；第二行为 4 张图片，并向这 4 张图片中添加 Invert 滤镜特效，实现底片效果；第三行和第一行的内容相同，都是向用户展示胶卷底片边框的内容。完整的表格代码如下：

```
<table cellpadding="0" cellspacing="0" border="0">
    <tr style="background-image:url(images/jiao1.jpg);
background-repeat:repeat-x">
        <td style="height:77px;"> </td>
    </tr>
    <tr>
        <td style="height:250px">
            <img src="images/dcca492a39a3aa7c5243c1bb.jpg"
style="filter:Invert()" width="200px" height="250px"/>
            <img src="images/124X0J32B03a2L.jpg" style="filter:Invert()"
width="200px" height="250px" />
            <img src="images/2400124.jpg" style="filter:Invert()"
width="200px" height="250px" />
            <img src="images/20064131117330.jpg" style="filter:Invert()"
width="200px" height="250px" />
        </td>
    </tr>
    <tr style="background-image:url(images/jiao.jpg);
background-repeat:repeat-x">
        <td style="height:96px;" valign="top"> </td>
    </tr>
</table>
```

(3) 保存页面。

14.10.4 运行结果

在浏览器中查看相册底片的效果，如图 14-23 所示。



图 14-23　相册底片效果

14.10.5　实例分析



源码解析：

在该实例中的四张人物图片都使用了 Invert 滤镜特效，Invert 滤镜能使图像达到底片的效果。其实，Invert 滤镜和 Xray 滤镜的使用不仅可以在图像标签<img>中添加“filter:Invert()/filter:Xray()”代码，还可以在页面元素所在的容器，比如 div 容器、td 容器中添加“filter:Invert()/filter:Xray()”代码。

14.11　实现镭射灯的效果

在昏暗的灯光下，一缕灯光照进来，那是一种什么样的场景呢？而有时候，客户的要求就是这样“无所不有”，不用担心，使用 CSS 中的 Light 滤镜配合 JavaScript 脚本就可以实现这样的效果。



视频教学：光盘/视频/14/14.11　实现镭射灯的效果.avi　　**长度：**4 分钟

14.11.1　基础知识——Light 滤镜

Light 滤镜可以产生类似于光照的效果，它可以控制多个光源照向一个元素对象，并可以调节亮度以及颜色。

Light 滤镜的语法格式如下：

```
{filter:Light(enabled=bEnabled)}
```

Light 滤镜是所有滤镜中最复杂的一个，而且还需要脚本语言 JavaScript 配合使用。对于已定义的 Light 滤镜属性，可以调用它的方法(Method)来设置或改变属性。这些方法如表 14-6 所示。

表 14-6　Light 滤镜使用的方法

方　法	说　明	方　法	说　明
AddAmbient	加入包围的光源	Changstrength	改变光源的强度
AddCone	加入锥形光源	Clear	清除所有的光源
AddPoint	加入点光源	MoveLight	移动光源
Changcolor	改变光的颜色	Changstrength	改变光源的强度

14.11.2　实例描述

假设现在需要描绘出镭射灯照在一束鲜花上的全过程，那么可以使用 Light 滤镜与 JavaScript 相结合，改变光源的位置。

14.11.3　实例应用

【例 14-10】实现镭射灯的效果

(1) 新建 HTML 文件，命名为 14-11.html。

(2) 在页面中创建一个两行一列的表格，第一行显示“镭射灯效果”5 个字，第二行引入一张图片，并设置该图片的 CSS 样式，实现 Light 滤镜特效。表格的完整代码如下所示：

```
<table>
      <tr>
         <td style="color:DarkBlue; font-weight:bold">
             动态光源效果
         </td>
      </tr>
      <tr>
         <td id="light" style="filter: light(); width: 200px">
            <img src="images/6bdbfd17ca1a762dc93d6dc2.jpg">
         </td>
      </tr>
</table>
```

(3) Light 滤镜必须与 JavaScript 配合使用。编辑 JavaScript 代码，实现光照的效果，而且随着鼠标的移动，灯光照射的方向也不相同，类似于镭射灯的效果。JavaScript 代码如下所示：

```
<script language="JavaScript">
<!--
   var g_numlights=0;
   // 调用设置光源函数
   window.onload=setlights;
   // 获得鼠标句柄
```

```
    light.onmousemove=mousehandler;
    //建立光源的集合
    function setlights(){
        light.filters[0].clear();
        light.filters[0].addcone(0,0,5,100,100,255,255,0,60,30);
    }
    // 捕捉鼠标的位置来移动光线焦点
    function mousehandler(){
        x=(window.event.x-80);
        y=(window.event.y-80);
        light.filters[0].movelight(0,x,y,5,1);
    }
-->
</script>
```

要实现光照的效果，JavaScript 脚本语言起了主要的作用。其中调用了滤镜的各种方法，首先要创建光源并指定光源位置。setlights 函数中的 filters[0]表示设置的光源滤镜，调用 clear 方法表示在每次页面加载时先清除所有的光源，然后再使用 addcone 方法创建锥形光源，该方法中的 10 个参数的意思为：第 1 个参数：光源的 X 坐标；第 2 个参数：光源的 Y 坐标；第 3 个参数：光源离开页面的高度；第 4～第 6 个参数：光源的颜色；第 7 个参数：光源光的色相；第 8 个参数：光源的形状；第 9 个参数：光源的光的强度；第 10 个参数：光源光散开的范围。如果需要在图片上添加多束光源，则可以重复使用 addcone 方法，但注意要使用不同的参数，否则光源处于同一位置，就无法产生效果了。函数 mousehandler 用来实现光束随着鼠标移动的效果。

14.11.4　运行结果

在浏览器中查看 Light 滤镜的效果，如图 14-24 和图 14-25 所示。

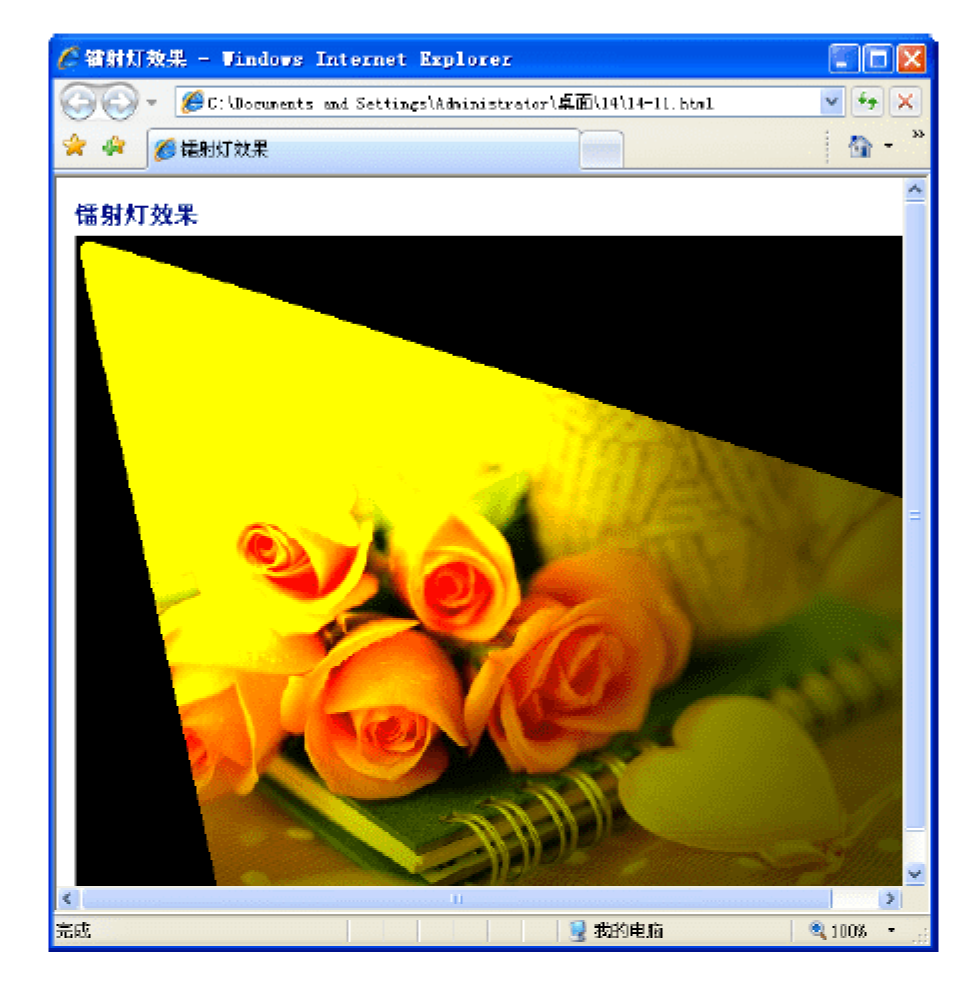


图 14-24　光源照射

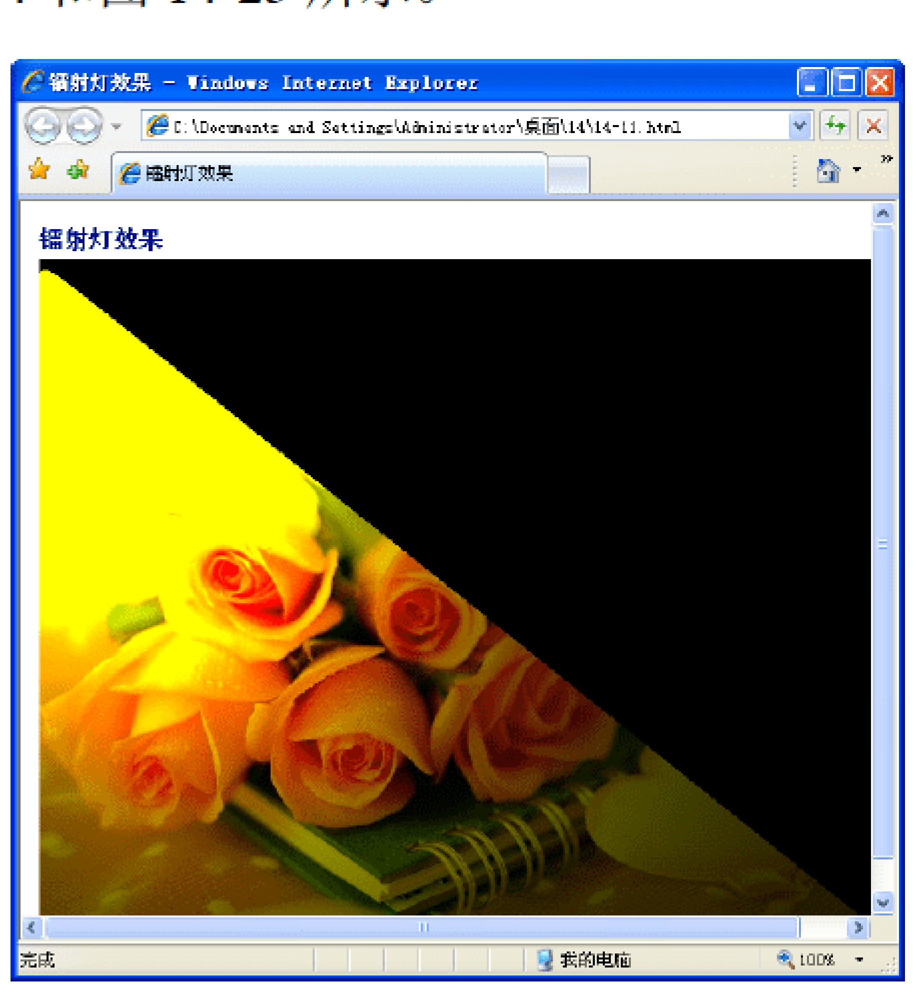


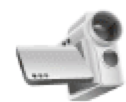
图 14-25　移动光束

14.11.5 实例分析


源码解析：
在这个实例中，使用 Light 滤镜的同时也使用了 JavaScript 脚本。JavaScript 脚本中，指定了窗口加载时执行函数 setlights()；并指定鼠标移动时执行的函数为 mousehandler()，也就是当鼠标移动时，激发这个函数，并执行它，从而实现锥形光束随着鼠标移动的效果。

14.12 制作遮罩字

所谓遮罩，就是使用一个颜色图层将包含文字或图像等对象的区域遮盖，但是对象却以背景色显示出来，这种效果在 Web 开发中很常用。使用 CSS 提供的 Mask 滤镜就可以达到这样的效果，本节将介绍 Mask 滤镜特效的使用方法。

 **视频教学：**光盘/视频/14/14.12 制作遮罩字.avi 长度：3 分钟

14.12.1 基础知识——Mask 滤镜

Mask 滤镜可以实现一种遮罩效果，所谓遮罩，就是使用一个颜色图层将包含有文字等对象的区域遮盖，但是文字或图像部分却以背景色显示出来。简单地说，该颜色图层就如同一张有颜色但透明的纸，它能够遮盖背景色却可以将对象的内容显示出来，但是内容的颜色由背景色所替代。



Mask 滤镜不能应用于图像文件以实现相应的效果。

Mask 滤镜的语法格式如下：

```
{filter:Mask(enabled=bEnabled , color=sColor)}
```

参数 color 用来设置 Mask 滤镜作用的颜色。

14.12.2 实例描述

“床前明月光，疑是地上霜”，这是一种很奇妙的景象，就连天上的玉兔和地上的白猫也来到了这户人家门前，它们很好奇地向屋里张望着，希望能看到些什么。于是，李白题词“床前明月光”。

14.12.3 实例应用

【例 14-11】制作遮罩字

(1) 新建 HTML 文件，命名为 14-12.html。

(2) 在页面中编辑“床前明月光”五个大字，并使用 Mask 滤镜特效，实现遮罩效果。关键代码如下所示：

```
<style type="text/css">
<!--
#apDiv4 {
    position:absolute;
    width:77px;
    height:368px;
    z-index:4;
    left: 15px;
    top: 17px;
    font-size:50pt;
    font-weight:bold;
    color:white;
    font-family:'华文行楷';
    filter:mask(color=yellow);
}
-->
</style>
<body style="background-color:black">
    <div id="apDiv4">床前明月光</div>
</body>
```

(3) 为了能使页面更加漂亮，可以添加一些层或其他布局来优化这个页面。

(4) 保存页面代码。

14.12.4　运行结果

在 IE 浏览器中运行 14-12.html 页面，查看 Mask 滤镜效果，如图 14-26 所示。

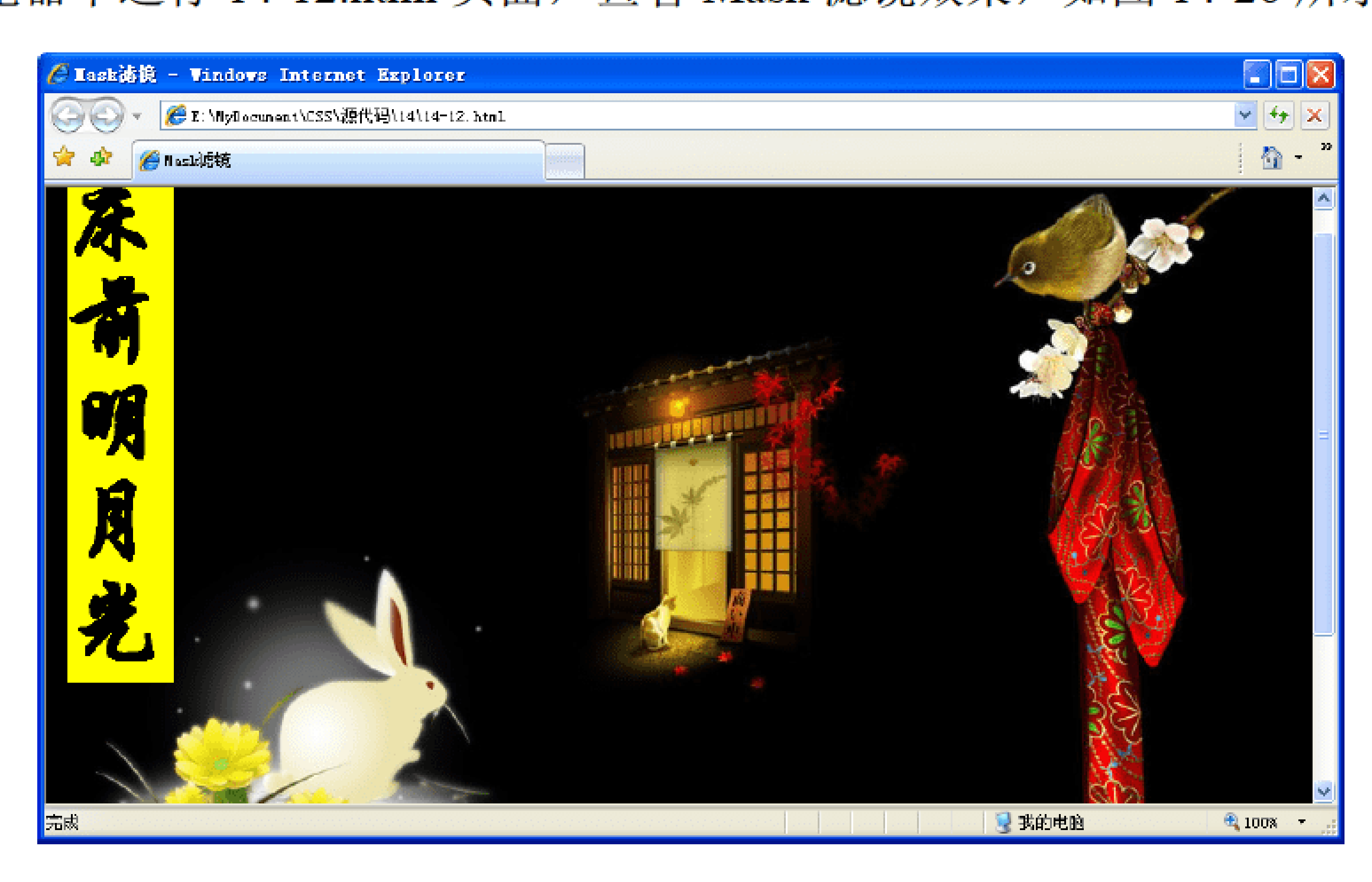


图 14-26　Mask 滤镜效果

14.12.5　实例分析


源码解析：

在这个实例中，为了显示需要的效果，设置背景色为黑色，以及字体颜色为白色，并设置Mask 滤镜作用的颜色为黄色，在浏览器中查看遮罩效果，如图 14-26 所示，“床前明月光”应用了 Mask 滤镜后，文字被滤镜设置的颜色覆盖，而且文字的颜色白色被背景色黑色所代替。

14.13　图像淡入淡出

前面介绍的 BlendTrans 滤镜能够实现图像的淡入淡出并相互转换的特效，本小节将要介绍的 RevealTrans 滤镜同样能够实现图像之间的切换效果。


视频教学：光盘/视频/14/14.13　图像淡入淡出.avi　　　　长度：5 分钟

14.13.1　基础知识——RevealTrans 滤镜

RevealTrans 滤镜能够产生 32 种动态效果，例如，溶解、水平(垂直)展幕、百叶窗等，而且还可以从这些动态效果中随机地选取其中一种效果进行切换。

RevealTrans 滤镜的语法格式如下：

```
filter : RevealTrans ( enabled=bEnabled, duration=fDuration,
transition=iTransitionType )
```

其中，enabled 表示是否激活该滤镜；duration 用于设置切换停留的时间；transition 用于指定转换方式，即指定要使用的动态效果，参数取值是 0～23。transition 参数值所代表的动态效果如表 14-7 所示。

表 14-7　RevealTrans 滤镜动态效果

动态效果	参 数 值	动态效果	参 数 值
矩形从大至小	0	随机溶解	12
矩形从小至大	1	从上下向中间展开	13
圆形从大至小	2	从中间向上下展开	14
圆形从小至大	3	从两边向中间展开	15
向上推开	4	从中间向两边展开	16
向下推开	5	从右上向左下展开	17
向右推开	6	从右下向左上展开	18
向左推开	7	从左上向右下展开	19

续表

动态效果	参 数 值	动态效果	参 数 值
垂直形百叶窗	8	从左下向右上展开	20
水平形百叶窗	9	随机水平细纹	21
水平棋盘	10	随机垂直细纹	22
垂直棋盘	11	随机选取一种效果	23

但是，如果只设置 transition 参数来实现切换过程，那么是不会有任何效果的，因为动态效果的实现还必须依靠脚本语言 JavaScript 调用相应的方法。

对于图片切换，实现代码与 BlendTrans 滤镜的转换代码相似，只不过定义类选择符样式规则变为

```
<style type="text/css">
  .revealtrans { filter:revealTrans(Transition=10,Duration=3)}
</style>
```

然后，在元素 img 中引用该样式规则即可，代码如下：

```
<img id="imgpic" class="revealtrans" src="25.jpg"/>
```

而 JavaScript 转换程序不用做任何修改，但是要注意一点，在<body>标签中需要添加页面加载事件，执行 playImg()函数，这样就完成了图片淡入淡出的效果。

14.13.2　实例描述

就如前面在讲述 BlendTrans 滤镜时所说的那样，几张图片轮换显示的效果是很常见的，那么，在学习 BlendTrans 滤镜之后，或许你感觉已经很神奇了，其实不然，RevealTrans 滤镜与 BlendTrans 滤镜相比更胜一筹。下面就来领略一下它的神奇之处吧！

14.13.3　实例应用

【例 14-12】两张图片渐入渐出轮换展示

(1) 新建 HTML 文件，命名为 RevealTrans.html。

(2) 在页面中定义实现 RevealTrans 滤镜特效的样式，CSS 样式代码如下：

```
    .revealtrans { filter:revealTrans(Transition=10,Duration=3)}
```

(3) 要使用 RevealTrans 滤镜特效实现两张图片轮换交替显示效果，需要结合 JavaScript 代码。编辑 JavaScript 代码，实现两张图片渐入渐出轮换展示的特效，JavaScript 代码如下所示：

```
<script language="JavaScript">
    //声明数组，数组元素的个数就是图片的个数，然后给数组元素赋值，值为图片路径
    imgNum=new ImgArray(2);
    imgNum[0]="images/67acb7a1387a65b046106437.jpg";
    imgNum[1]="images/5243525_70.jpg";
```

```
    //获取数组记录数
    function ImgArray(len){
        this.length=len;
    }
    var i=1;
    //转换过程
    function playImg()
    {
        if(i==1)
        {
            i=0;
        }
        else{
            i++;
        }
        imgpic.filters[0].apply();
        imgpic.src=imgNum[i];
        imgpic.filters[0].play();
        timeout=setTimeout('playImg()',4000);
    }
</script>
```

(4) 在页面中引入一张图片，作为初始化页面时展示的图片，并在<body>标签中定义“onload”事件，执行的函数为上面 JavaScript 代码中定义的 playImg()函数。代码如下：

```
<body onload="playImg()">
    <img src="images/3.jpg"  class="revealtrans" id="imgpic"/>
</body>
```

14.13.4 运行结果

在浏览器中查看切换动态效果，如图 14-27～图 14-29 所示。

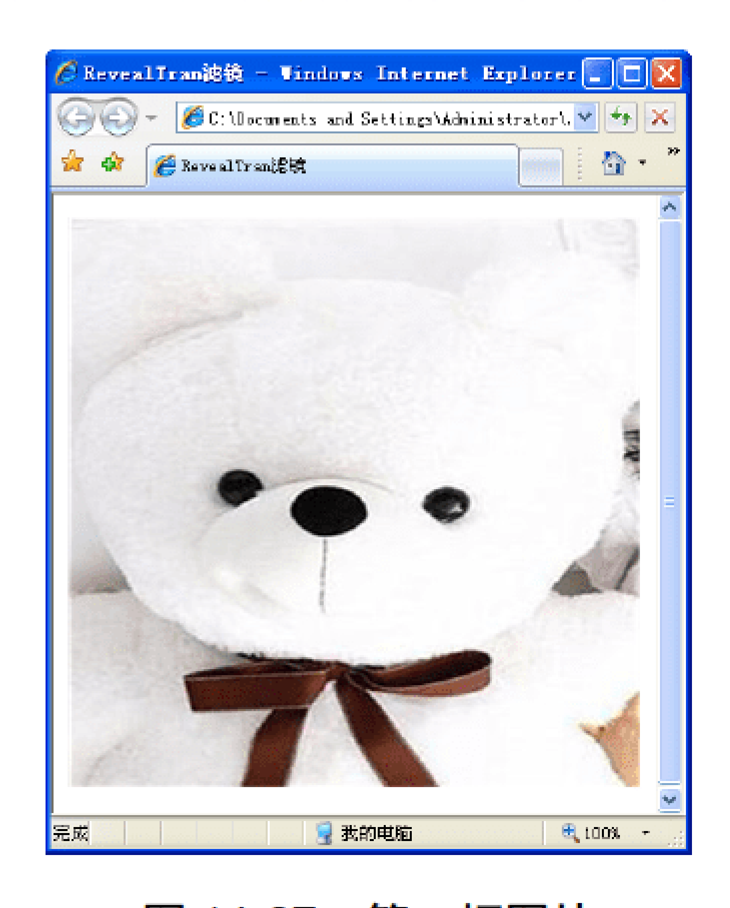


图 14-27 第一幅图片

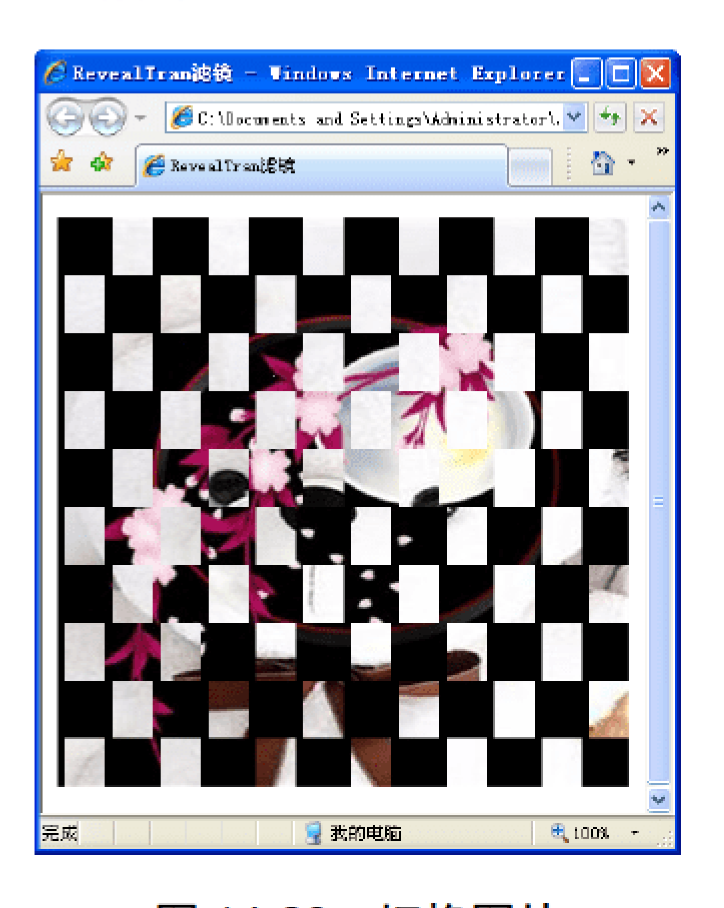


图 14-28 切换图片

图 14-29　第二幅图片

14.13.5　实例分析



源码解析：

从这个实例中，可以看出 RevealTrans 滤镜和 BlendTrans 滤镜实现的效果类似，实现代码也相似，只需要把类选择符的样式规则改变一下就能完成不同的滤镜效果，其他的都不需要改变。

14.14　让图像产生波纹变形

使用 CSS 中的 Wave 滤镜可使页面对象产生波纹变形，给人一种混淆的感觉。本节将为大家介绍 Wave 滤镜的使用。



视频教学：光盘/视频/14/14.14　让图像产生波纹变形.avi　　长度：14 分钟

14.14.1　基础知识——Wave 滤镜

Wave 滤镜可使页面对象产生波纹变形，其语法格式如下：

```
{filter:Wave(enabled=bEnabled,add=bAddImage,freq=iWaveCount,
lightStrength=iPercentage,phase=iPercentage,strength=iDistance)}
```

各参数说明如表 14-8 所示。

表 14-8　Wave 滤镜参数

参　数	说　明
Enabled	设置滤镜是否激活
Add	布尔值，表示是否在原始对象上显示效果。True 表示显示；False 表示不显示
Freq	设置生成波纹的频率，也就是设定在对象上产生的完整的波纹的条数
lightStrength	波纹效果的光照强度，取值 0～100
Phase	设置正弦波开始的偏移量，取百分比值 0～100，默认值为 0。25 就是 360×25%为 90 度，50 则为 180 度
Strength	波纹的曲折的强度

无论在图片还是文字上应用 Wave 滤镜，都能实现波纹的效果，实例代码如下：

```
原图<img src="1403" width="300" height="252" />
图一<img style="filter:wave(add=false,freq=3,lightstrength=20,phase=0,strength=5)"
src="1403.jpg" width="300" height="252" />
图二<img style="filter:wave(add=false,freq=10,lightstrength=20,phase=45,strength=10)"
src="1403.jpg" width="300" height="252" />
<div style="filter:wave(add=true,freq=3,lightstrength=100,phase=45,strength=20);">
    Happly New Year!
</div>
```

代码中设置文字 Wave 滤镜中参数 add 值为 true，表示应用滤镜效果的文字与未应用的文字一同显示。在浏览器中查看效果，如图 14-30 所示。



图 14-30　Wave 滤镜效果

14.14.2　实例描述

Wave 滤镜可以给人一种动态的效果。下面我们用 Wave 滤镜特效来制作一张贺卡。

14.14.3　实例应用

【例 14-13】制作波纹变形的新年贺卡

(1) 新建 HTML 文件，命名为 14-13.html。

(2) 在页面中使用 DIV 布局，显示新年贺卡页面，页面布局代码如下所示：

```
<div id="apDiv1"><img src="images/20081218211217271.jpg"/></div>
<div id="apDiv2"><img src="images/20071128172813677_2.jpg"/>
<div id="apDiv3">
    <img
style="filter:wave(add=false,freq=3,lightstrength=20,phase=30,strength=5)"
src="images/untitled.jpg" width="128" height="128" />
</div>
<div id="apDiv4">
   <p>恭</p>
   <p>贺</p>
   <p>新</p>
   <p>年</p>
  </div>
</div>
```

(3) 编辑贺卡的 CSS 样式代码，使“apDiv4”层产生波纹特效，代码如下：

```
#apDiv1 {
    position:absolute;
    width:168px;
    height:159px;
    z-index:1;
    left: 28px;
    top: 69px;
}
#apDiv2 {
    position:absolute;
    width:200px;
    height:780px;
    z-index:2;
    left: 12px;
    top: 11px;
}
#apDiv3 {
    position:absolute;
    width:132px;
    height:132px;
```

```
    z-index:1;
    left: 154px;
    top: 248px;
}
#apDiv4 {
    position:absolute;
    width:422px;
    height:637px;
    z-index:2;
    left: 421px;
    top: 7px;
    background-image:url(images/he.jpg);
    font-size:70pt;
    font-weight:bold;
    text-align:center;
    color:#FFFF00;
    filter:wave(add=true,freq=3,lightstrength=100,phase=45,strength=20);
}
```

(4) 保存页面。

14.14.4 运行结果

在浏览器中运行 14-13.html 页面的效果如图 14-31 所示。



图 14-31 产生波纹变形的新年贺卡

14.14.5　实例分析

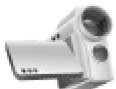


源码解析：

在这个实例中，定义了三个层，分别是 apDiv2 层、apDiv3 层和 apDiv2 层。其中 apDiv2 层中容纳了贺卡的背景图片，同时也使用了 Wave 滤镜特效，使贺卡给人一种动态感；apDiv3 中容纳了“小熊贺岁”的图片，并使用了 Wave 滤镜特效；apDiv4 中容纳了“恭贺新年”四个字，也使用了 Wave 滤镜特效，也就是说上面的案例中有 3 处用到了 Wave 滤镜特效，并定义了正弦波偏移量分别为 45、30、45。在 apDiv4 中的文字使用 Wave 滤镜时，指定滤镜中参数 add 值为 True，表示应用滤镜效果的文字与未应用的文字一同显示。

14.15　鼠 标 指 针

经常使用操作系统的人们都知道，当把鼠标移动到不同的地方，或执行不同的操作时，鼠标的样式是不同的。比如，当需要伸缩窗口时，将鼠标放置在窗口边沿处，鼠标会变成双向箭头状↔；当系统繁忙时，鼠标会变成漏斗状⌛。CSS 中提供了能实现对鼠标样式控制的属性，本节将介绍如何使用 CSS 样式控制鼠标指针。



视频教学：光盘/视频/14/14.15　鼠标指针.avi　　　　**长度：**3 分钟

14.15.1　基础知识——鼠标指针

CSS 不仅能够准确地控制及美化页面，而且还能定义鼠标指针的样式。当鼠标移动到不同的元素对象上面时，使鼠标以不同的形状或图像显示。CSS 通过改变 cursor 属性(鼠标指针属性)来实现对鼠标样式的控制。

cursor 属性包含 17 个属性值，对应鼠标的 17 种样式，而且还能够通过 url 的链接地址自定义鼠标指针，如表 14-9 所示。

表 14-9　鼠标样式

属性值	说　明	鼠标样式	属性值	说　明	鼠标样式
auto	自动，按照默认状态自行改变		s-resize	箭头朝下双向	↕
crosshair	精确定位十字	＋	w-resize	箭头朝左双向	↔
default	默认鼠标指针	↖	e-resize	箭头朝右双向	↔
hand	手形	☝	ne-resize	箭头右上双向	⤢
move	移动	✥	se-resize	箭头右下双向	⤡

续表

属性值	说　明	鼠标样式	属性值	说　明	鼠标样式
help	帮助	↖?	nw-resize	箭头左上双向	↖
wait	等待	⌛	sw-resize	箭头左下双向	↙
text	文本	I	pointer	指示	☝
n-resize	箭头朝上双向	↕	url (url)	自定义鼠标指针	

14.15.2　实例描述

美图秀秀是一款很好用的国产免费图片处理软件，软件的操作和程序相对于专业图片处理软件如光影魔术手、Photoshop 比较简单，因此它很受大家的喜爱。用过美图秀秀的同学还记得把鼠标放在要编辑的图片上时的样式吗？需要给图片加一个配饰，把鼠标放在配饰上时的样式是怎样的呢？下面就来做一个仿美图秀秀主页的页面。

14.15.3　实例应用

【例 14-14】制作美图秀秀主页

(1) 新建 HTML 文件，命名为 14-14.html。

(2) 在页面中有两个区域，一个是图片主区域，另一个是饰品区域。在这两个区域中添加鼠标样式，这两个区域采用 div 布局来实现，代码如下：

```
<div id="apDiv3">
    <div id="apDiv5"><img src="images/mei5.jpg"/></div>
    <img src="images/mei3.jpg" style="cursor:wait"/>
</div>
<div id="apDiv4">
  <img src="images/mei4.jpg" width="200" height="448" />
  <div id="apDiv7"><img src="images/mei6.jpg" width="83" height="75"
style="cursor:pointer;"/></div>
</div>
<div id="apDiv6">
    <img src="images/20081218211217271.jpg" width="200" height="250"
style="cursor:crosshair"/>
</div>
```

(3) 在页面中添加响应的 CSS 样式，使之与美图秀秀中的主页相似。

(4) 保存 14-14.html 页面。

14.15.4　运行结果

在浏览器中查看鼠标样式，如图 14-32～图 14-34 所示。

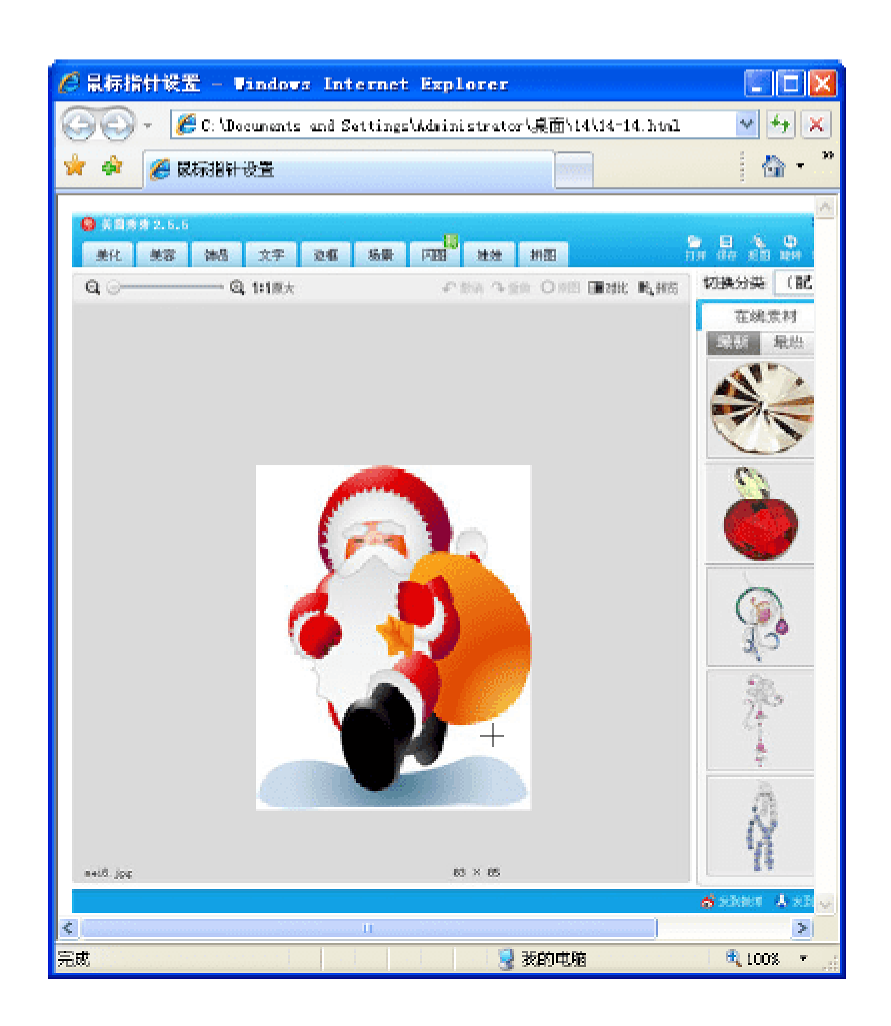


图 14-32　精确定位十字鼠标样式

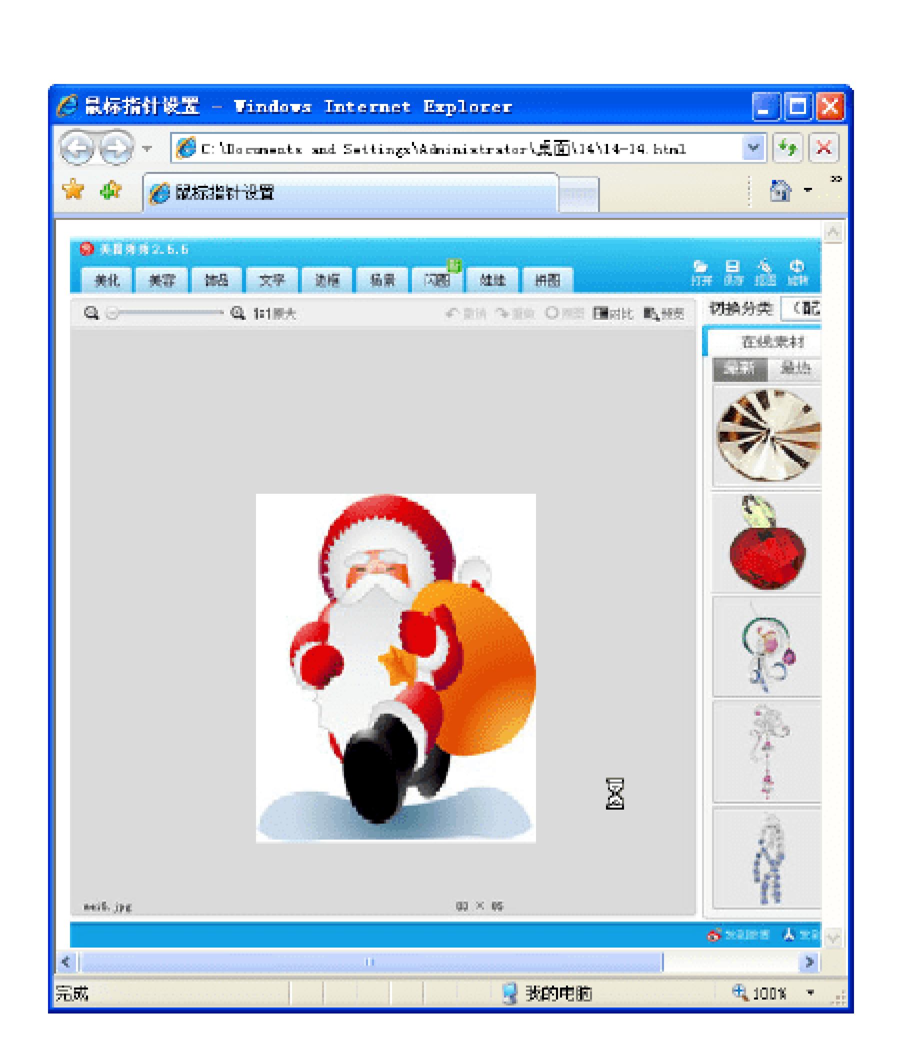


图 14-33　等待鼠标样式

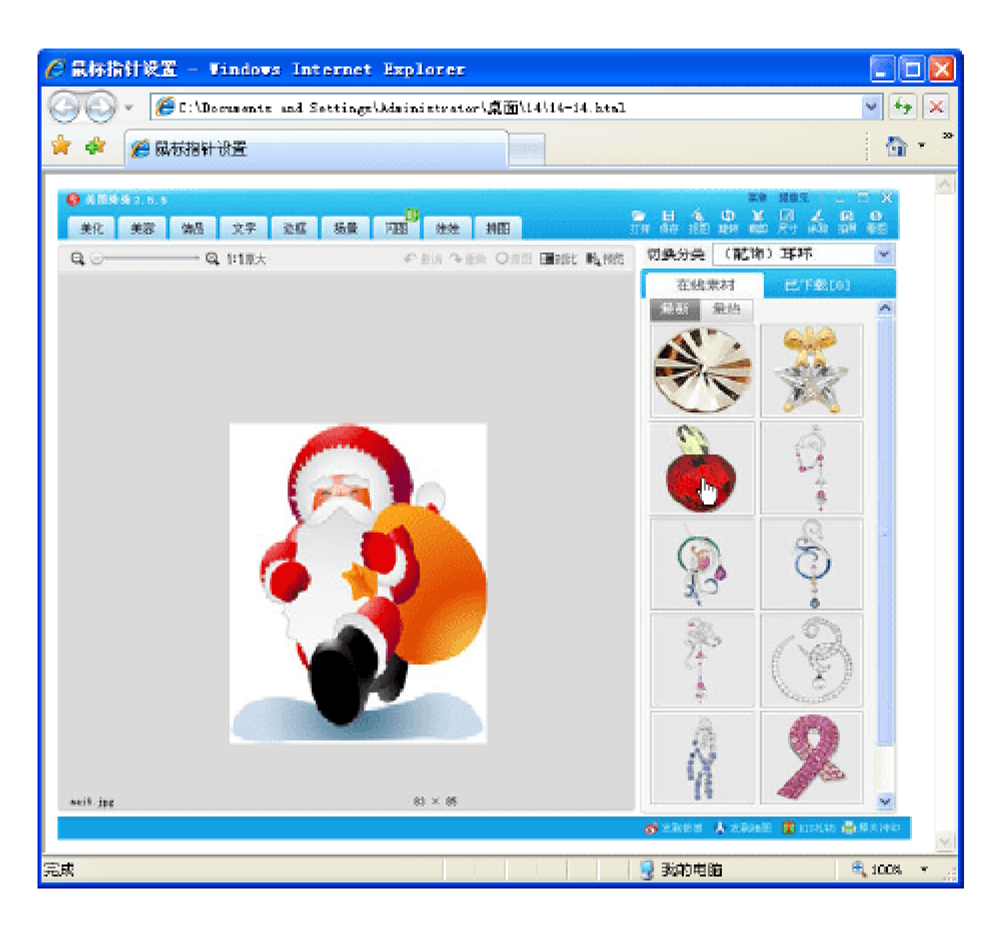


图 14-34　手形鼠标样式

14.15.5　实例分析



源码解析：

在这个实例中，当鼠标移动到人物图像上时，样式为一个“十字形”；当鼠标移动到图像主区域上时，样式为“沙漏”；当鼠标移动到配饰上时，样式为“手形”，这些都只需要改变 CSS 中的 cursor 属性即可。这里需要重点说明的是自定义鼠标样式，为使页面更加美观，许多网站使用了各种样式的鼠标指针来代替传统的鼠标样式。鼠标指针分为静态鼠标指针(.cur)和动态鼠标指针(.ani)两类，所以自定义鼠标样式中 url 引用的鼠标指针的扩展名称是.cur 或.ani，而且括号内的名称不需要引号，即 cursor:url(Arrow.cur)。

14.16 常见问题解答

14.16.1 对一个对象同时使用多个滤镜的问题



对一个对象同时使用多个滤镜的问题。
网络课堂：http://bbs.itzcn.com/thread-12267-1-1.html

最近闲来无事，就随便做了个使用滤镜特效的页面，但是，在页面中我对一个对象同时使用了多个滤镜，却不起作用，怎么回事啊？对一个对象同时使用多个滤镜的语法是什么啊？

【解决办法】不明白如何使用滤镜集合，通俗点讲就是不知道如何对一个对象同时使用多个滤镜。其实很简单，它的语法就是：

```
filter:滤镜 1 滤镜 2 滤镜 3……
```

一定要记住滤镜之间有空格，不要写成：

```
filter:滤镜 1filter:滤镜 2filter:滤镜 3……
```

或者

```
filter:滤镜 1;滤镜 2;滤镜 3……
```

14.16.2 startx、starty、finishx、finishy 参数是否重要



startx、starty、finishx、finishy 参数重要吗
网络课堂：http://bbs.itzcn.com/thread-12268-1-1.html

好多 CSS 教程中都没有具体讲解 startx、starty、finishx、finishy 这两对参数，这些参数重要吗？使用这些参数可以实现什么样的效果呢？

【解决办法】CSS 滤镜中的 startx、starty、finishx、finishy 参数对于控制部分滤镜的作用范围是非常重要的，而对于这两对参数其后所跟的数值众说纷纭，初学者可能无法分辨，在没有任何前置条件限制下，该参数应该是滤镜作用对象的横纵坐标的百分比值。大家可以看下面的这个例子理解这两对参数的实现方法，这是一个简单的渐变效果代码：

```
<div style="width:800px; height:200px; filter :
progid:DXImageTransform.Microsoft.Alpha(style=1,opacity=10,finishOpacity=10
0,startx=0,starty=0,finishx=0, finishy=35)
progid:DXImageTransform.Microsoft.Alpha(style=1,opacity=50,finishOpacity=50
,startx=0,starty=35,finishx=0, finishy=65)
progid:DXImageTransform.Microsoft.Alpha(style=1,opacity=100,finishOpacity=1
0,startx=0,starty=65,finishx=0, finishy=100); background:#0033FF;" >
</div>
```

14.16.3　IE6 中使用 CSS 滤镜后链接无法单击的问题



IE6 中使用 CSS 滤镜后链接无法单击的问题。
网络课堂：http://bbs.itzcn.com/thread-12270-1-1.html

网上最大众化地解决 IE6 无法显示 png 背景图片透明问题的方法就是用滤镜实现的，代码如下：

```
filter: progid:DXImageTransform.Microsoft.AlphaImageLoader(enabled=true,
sizingMethod=scale, src='images/liBack.png',no-repeat);
background:none;
```

而用了上面的代码之后会导致链接失效，这是为什么啊？

【解决办法】这个问题想必很多人都会遇到，其实解决办法很简单，只需要给链接加上这样的代码即可：

```
position:relative
```

如下面的代码：

```
* html #nav li{
    filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(enabled=true,
sizingMethod=scale, src='images/liBack.png',no-repeat); background:none;
}
#nav li a{
    position:relative;
}
```

14.17　习　　题

1. 填空题

(1) ＿＿＿＿＿滤镜是将页面元素与背景混合，从而产生透明渐变的效果。而且可以通过改变数值来控制透明度。如果将 Alpha 滤镜与网页脚本语言结合，并适当地设置其参数，就能使图像显示淡入淡出的效果。

(2) Glow 滤镜的语法格式如下：

```
{filter : Glow ( enabled=bEnabled , color=sColor , strength=iDistance ) }
```

其中，参数 color 设置＿＿＿＿＿的颜色，strength 设置了晕圈的范围，值范围是 1～255，值越大效果越强。

(3) 假设一张图像需要实现光照的效果，犹如手电筒照在这张图像上，需要使用

________滤镜。

(4) 在页面中实现仿 Flash 播放多张图像动画的效果，使图像以溶解、水平(垂直)展幕、百叶窗等样式显示与隐藏，需要使用________滤镜。

2. 选择题

(1) Alpha 滤镜中的 style 参数设置透明渐变的样式，也就是渐变显示的形状，取值为 0～3。其中________表示圆形渐变。

A. 0　　B. 1　　C. 2　　D. 3

(2) ________滤镜用于使页面元素产生模糊效果，甚至可以产生高速移动的动感效果。

A. Blur　　B. Shadow　　C. Invert　　D. Mask

(3) 下面描述正确的是________。

A. Shadow 滤镜和 DropShadow 滤镜不同的是，Shadow 滤镜能够在指定的方向上设置投影，且投影可产生渐进的效果，而 DropShadow 滤镜是利用偏移量来定义投影位置。

B. Invert 滤镜可以设置对象中指定的颜色为透明色。在页面元素中将该滤镜应用于对象时，则对象内容中的指定颜色就变为透明的，即不可见了。

C. FlipV 滤镜用于水平翻转对象，即将元素对象按水平方向进行 180 度翻转。

D. Xray 滤镜一般应用于图像，用来将一幅图像转变为灰度图，即图像中的色彩全部由黑、白、灰三种颜色所代替。

(4) 假设 1404.jpg 图片上有一朵粉色的花朵，那么使用 Gray()滤镜之后，如下面的代码：

```
<img src="images/1404.jpg" width="240" height="210"
style="filter:Gray()"/>
```

在页面上输出的图片花朵颜色是________。

A. 让 1404.jpg 图像反映出它的轮廓并将这些轮廓加亮，产生的效果类似于 X 光片

B. 花朵背景由白色变为灰色，花朵变为白色

C. 灰色

D. 粉色，不变

3. 上机练习

上机练习：设计椰子树水中倒影效果

使一张椰子树的图片在页面的一个层中出现 4 次，存放顺序是上面两张，下面两张，形成一个正方形。右上角的图片水平翻转，左下角的图片垂直翻转，右下角的图片水平、垂直同时翻转形成一棵在水中倒影的椰子树，如图 14-35 所示。



图 14-35　椰子树水中倒影效果



第 15 章　使用 Dreamweaver 提高工作效率

内容摘要：

随着计算机的逐渐普及，越来越多的人开始将网络作为一种娱乐消遣的平台。娱乐类的网站即是通过网络这个平台，给用户提供各种娱乐服务的网站，要想让网站视觉效果更加美观，在这里我首先推荐使用 Dreamweaver 进行布局编写页面。

Dreamweaver 是一个专业的 HTML 编辑器，用于对 Web 站点、Web 网页和 Web 应用程序进行设计、编码与开发。无论使用者喜欢直接编写 HTML 代码，还是偏爱在可视化编辑环境中工作，Dreamweaver 都能够提供众多的帮助工具，丰富使用者的网页创作体验。

学习目标：

- 学会使用 CSS 面板创建样式的方法
- 掌握保存 CSS 文件到本地的方法
- 熟悉在面板中添加 CSS 的背景属性
- 掌握制作二级导航菜单的方法
- 掌握制作图片展示的方法
- 掌握制作常识问答方法
- 掌握制作 Spry 表单方法

15.1 使用 Dreamweaver 完成网站设计

在网页开发领域，CSS 页面布局占据了很大的分量。为了方便创建一个合理布局的页面，可以使用 Dreamweaver。这节就通过 Dreamweaver 开发工具来创建 CSS 样式信息。

**视频教学：**光盘/视频/15/15.1 使用 Dreamweaver 完成网站介绍.avi　　长度：10 分钟

15.1.1 基础知识——使用 Dreamweaver 编写 HTML 信息

HTML 是一种简单易学的标记语言，分为文档头与文档体两部分。在文档头中对这个文档进行一些必要的定义，文档体中是要显示的各种文档信息。最简单的网页 HTML 代码如下：

```
<html>
<head>
<title>首页</title>
</head>
<body>
正文内容
</body>
</html>
```

其中<html></html>在最外层，表示这对标记间的内容是 HTML 文档；<head></head>之间包括文档的头部信息，比如文档标题；<body></body>之间是正文内容部分。

Dreamweaver 是一个所见即所得的编辑器，对 HTML 代码具有非常强大的编辑功能。它允许我们在代码检查器、代码视图与拆分视图中分别进行查看与编辑。

1. 代码检查器

Dreamweaver 中的代码检查器是专门用来查看与编辑 HTML 代码的，执行“窗口”|“代码检查器”命令(快捷键 F10)，打开“代码检查器”对话框，其中包括了各种编码工具，如图 15-1 所示。

图 15-1 则是使用 Dreamweaver 工具编写代码的面板，同样还可以在这个面板内进行修改、添加和删除 CSS 样式信息等。编码工具栏中的各种工具名称及功能简介如表 15-1 所示。



图 15-1　“代码检查器”对话框

表 15-1　编码工具栏中的各种工具名称及其作用

名　称	图标	作　用
打开文档		单击该按钮显示代码所在文件的路径
折叠整个标签		单击该按钮将所有代码折叠在一起，如果按住 Alt 键单击，则可以折叠外部标签
折叠所选		单击该按钮将选中的代码折叠在一起，如果按住 Alt 键单击，则可以折叠外部标签
扩展全部		单击该按钮可以将所有折叠的代码展开
选择父标签		单击该按钮则将<html>与</html>之间的代码选中
选取当前代码段		单击该按钮选中鼠标所在的 JavaScript 代码段
行号		单击该按钮显示或者隐藏代码视图中的代码行号
高亮显示无效代码		启用该按钮则可以显示代码中的无效代码
应用注释		单击该按钮后弹出菜单，其中“应用 HTML 注释”命令为静态网页代码中的注释标签，“应用/* */注释”命令应用于 CSS 样式中的注释，“应用//注释”命令应用于 JavaScript 代码中的注释，“应用’注释”命令为动态网页代码中的注释标签
删除注释		选中代码中的注释后单击该按钮，则会将注释标签删除
环绕标签		选中代码单击该按钮，则为其添加围堵标签。单击该按钮后在代码视图中会显示围堵标签列表，从中可以选择要添加的标签
最近的代码片段		单击该按钮，在弹出的菜单中会显示最近使用的 JavaScript 代码名称，以及“代码片段”面板命令
缩进代码		单击该按钮可以将光标所在的代码缩进，使其具有递进关系
凸出代码		单击该按钮可以将光标所在的代码向前凸出
格式化源代码		单击该按钮弹出菜单，其中“套用源格式”命令是将选中的代码以标准格式显示，而“代码格式设置”和“编辑标签库”命令则可以编辑代码格式

编码工具栏包含可用于执行多种标准编码操作的按钮，例如折叠和展开所选代码、应用和删除注释、缩进代码、插入最近使用过的代码片段等。下面介绍如何使用“缩进代码”按钮使 HTML 代码具有递进关系，首先将光标放置在要缩进的代码左侧或者选中该代码，然后单击“缩进代码”按钮，如图 15-2 所示，即可将代码向右缩进，如图 15-3 所示。

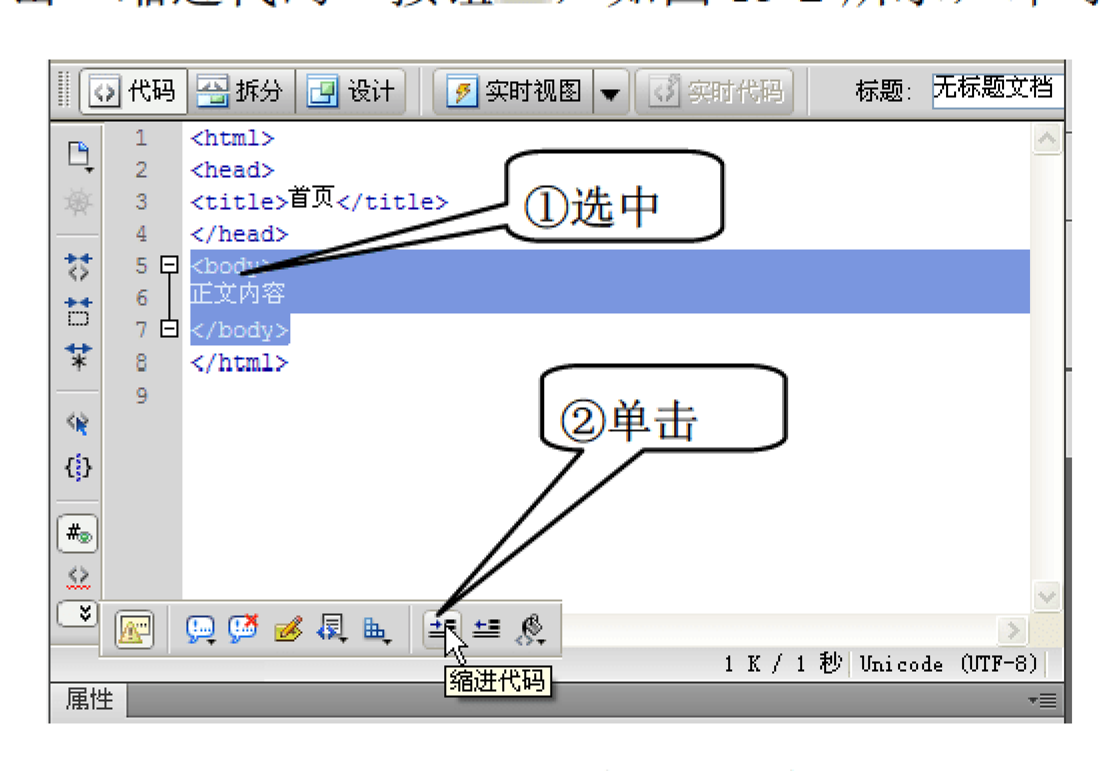


图 15-2　选择缩进代码按钮

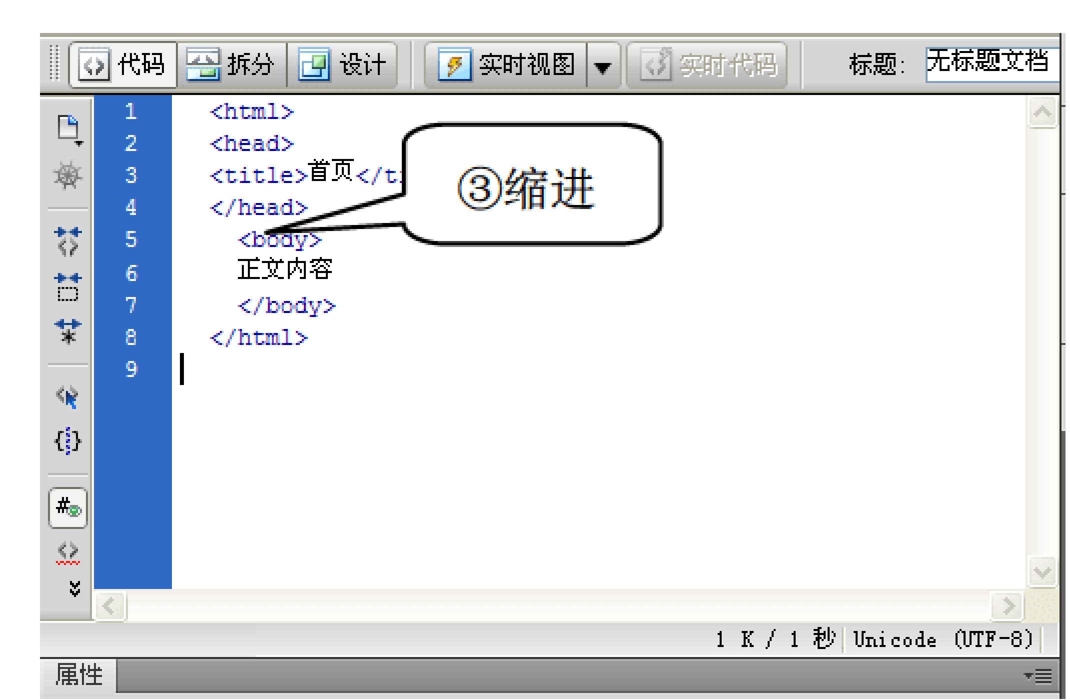


图 15-3　缩进后的代码



代码缩进与设计视图中的文本缩进操作方法相同。

2．代码视图与拆分视图

Dreamweaver 文档编辑窗口中有三种查看文档的方式，分别为代码视图、拆分视图和设计视图。分别单击不同的视图按钮，窗口中就会显示相应的编辑视图，其中代码视图与拆分视图均可以查看与编辑 HTML 代码。

单击文档工具栏上的“显示代码视图”按钮代码，文档窗口就切换到代码视图状态，如图 15-4 所示。发现编码工具栏也放置在代码视图左侧，所以同样可以在该视图中编辑代码。

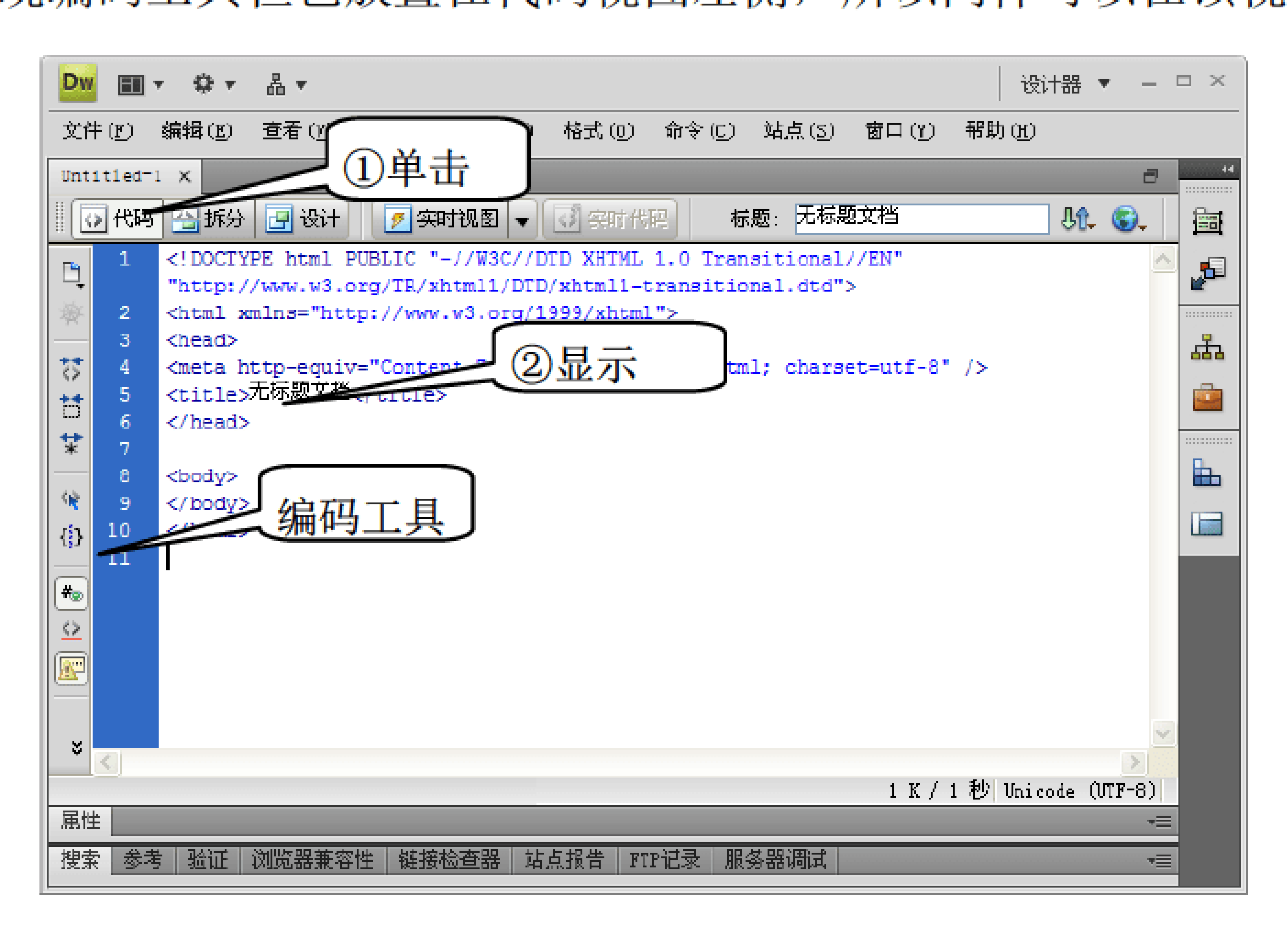


图 15-4　代码视图

单击文档工具栏上的“显示代码视图和设计视图”按钮拆分，文档窗口就会切换到拆分

视图状态，如图 15-5 所示。这样在设计视图中编辑网页元素，同时会在代码视图中看到相应的变化。



图 15-5　拆分视图

15.1.2　基础知识——使用 CSS 面板创建样式

在页面内创建 CSS 样式的方法有很多种，例如说直接编写和自动生成等多种方式，这节主要介绍如何使用 CSS 面板来创建 CSS 样式。

首先打开 Dreamweaver 文档编辑器，选择 Dreamweaver 工具菜单栏中的“窗口”|“CSS 样式”命令，此时在右侧将会显示如图 15-6 所示的面板。

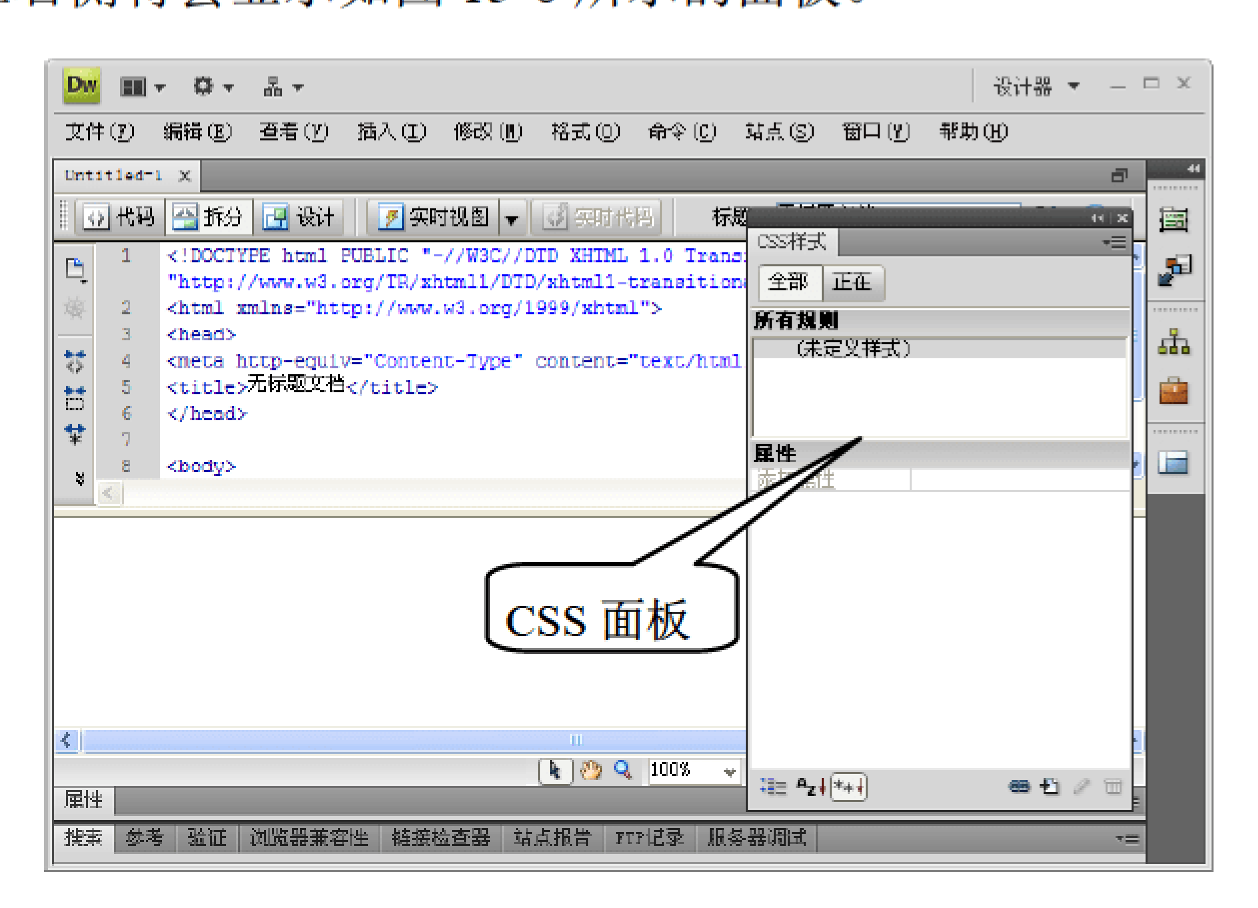


图 15-6　打开 CSS 面板

接下来单击 CSS 样式面板右下角的“新建 CSS 规则”按钮，此时将会弹出“新建 CSS 规则”对话框，详细结果如图 15-7 所示。

在“新建 CSS 规则”对话框内有“选择器类型”、“选择器名称”和“规则定义”3 个下拉列表框。

- “选择器类型”下拉列表框中有“类”、“ID”、“标签”和“符合内容”等选项，通过设置不同的选项可以定义 CSS 样式的类型规则。
- “选择器名称”下拉列表框中默认是没有值的，它是用来定义 CSS 样式的选择器名称的。在这里可以对它进行编辑，同样当选择“选择器类型”下拉列表框中的“标签”时，该下拉列表框中将会显示很多种标签的名称以供选择。
- “规则定义”下拉列表框内有“仅限该文档”和“新建样式表文件”两个选项，主要用来保存定义规则的位置。

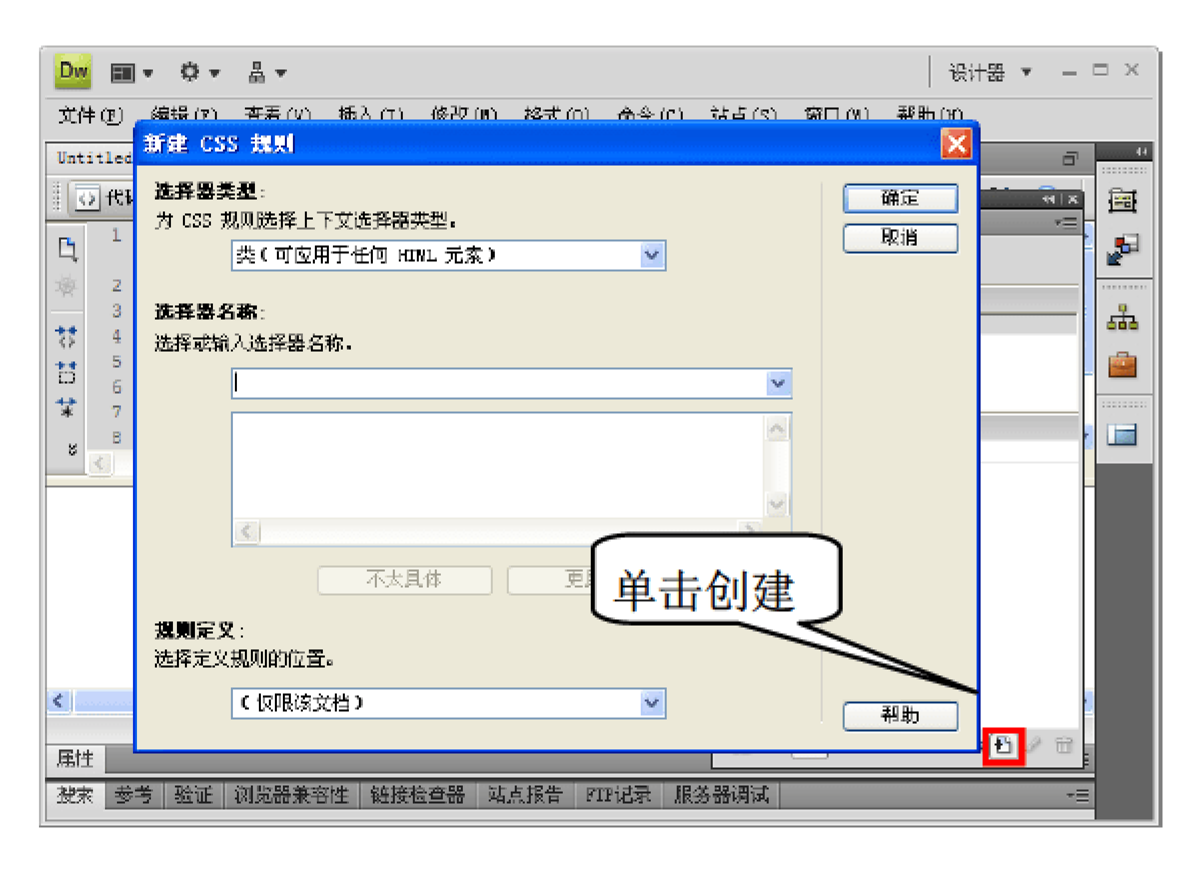


图 15-7 “新建 CSS 规则”对话框

通过设置以上 3 个选项的相关信息可以对 CSS 规则进行定义，首先设置“选择器类型”下拉列表框选项为“标签”，接下来设置“选择器名称”下拉列表框选项为“div”，最后设置“规则定义”下拉列表框的选项为“仅限该文档”，如图 15-8 所示。

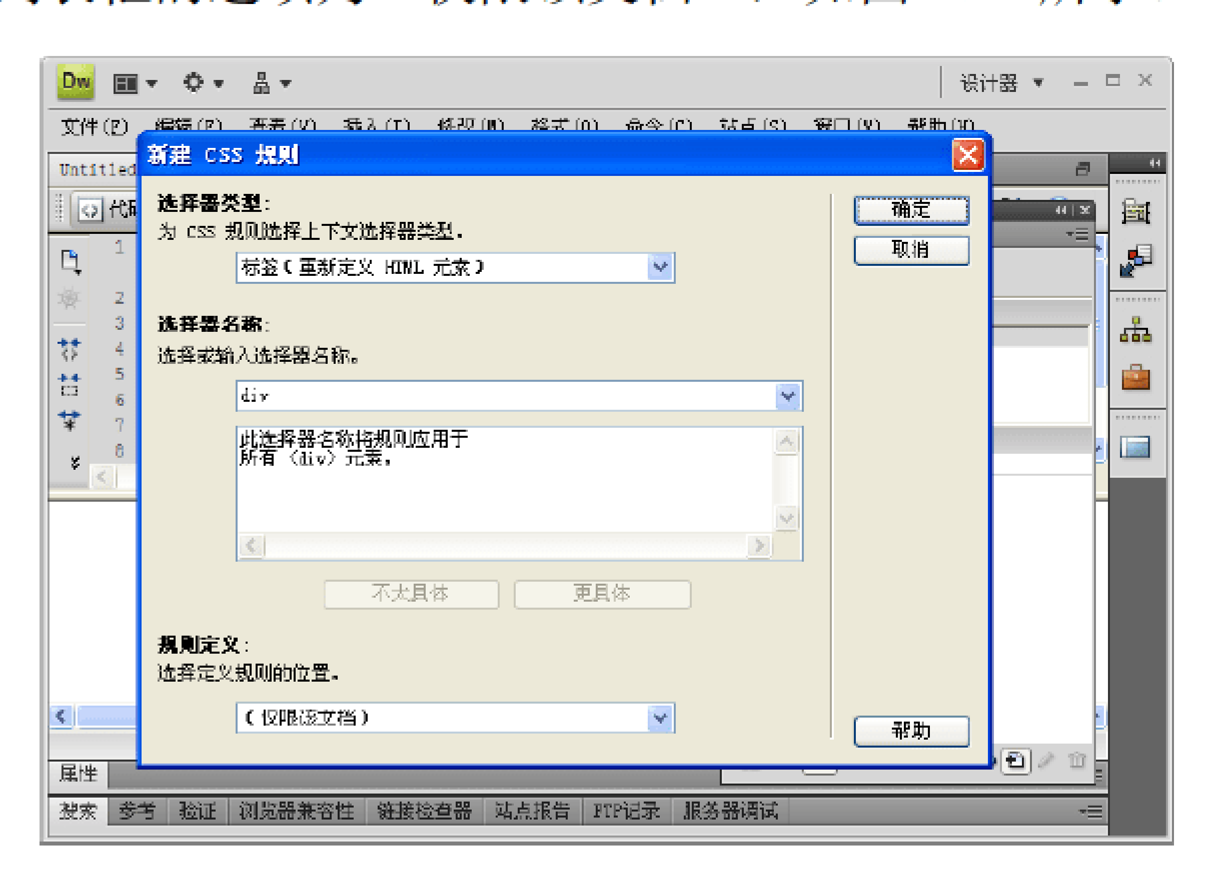


图 15-8 定义 CSS 规则

此时，设置完成后单击“确定”按钮，将会弹出“div 的 CSS 规则定义”对话框，如图 15-9 所示。

“div 的 CSS 规则定义”对话框由“分类”和“分类详细信息模块”两部分组成，其中在“分类”项中有类型、背景、区块、方框、边框、列表、定位和扩展等选项。通过选择“分类”中的各项，可以对该项中的详细信息进行设置，如图 15-10 所示。

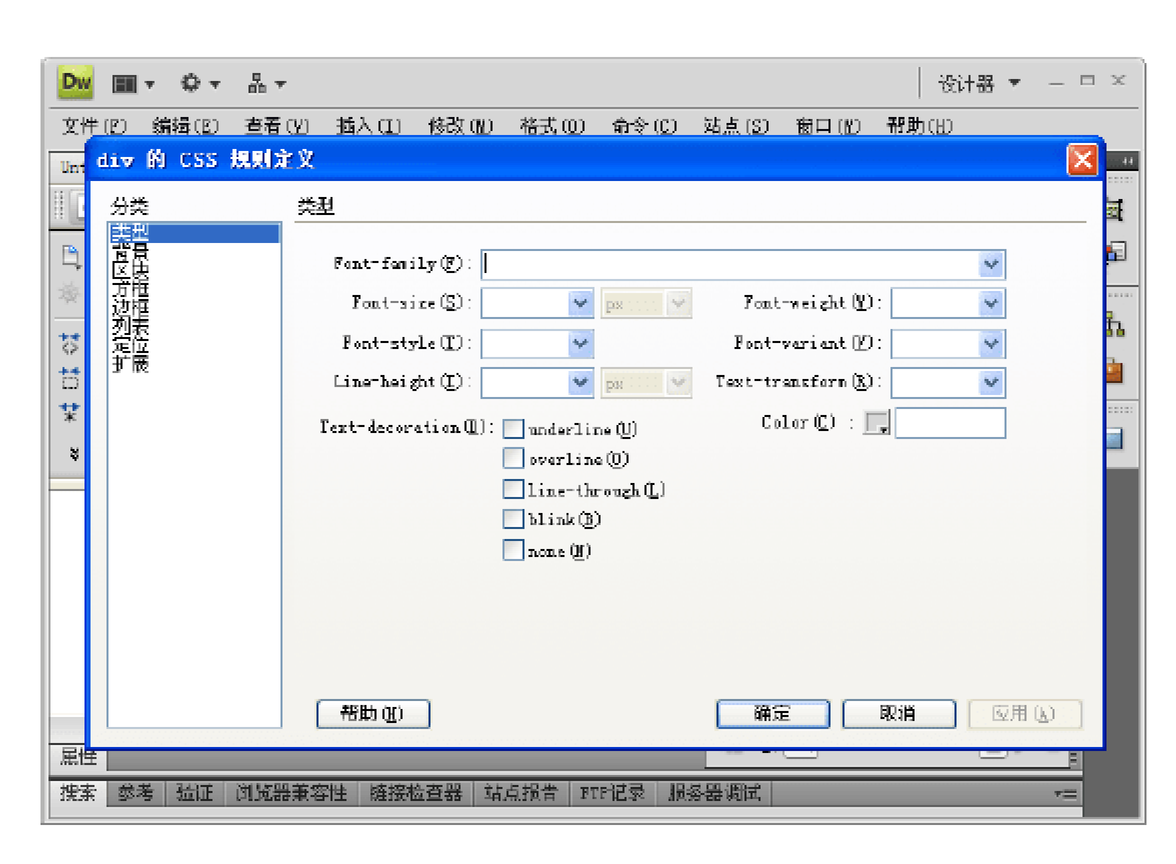


图 15-9　“div 的 CSS 规则定义”对话框

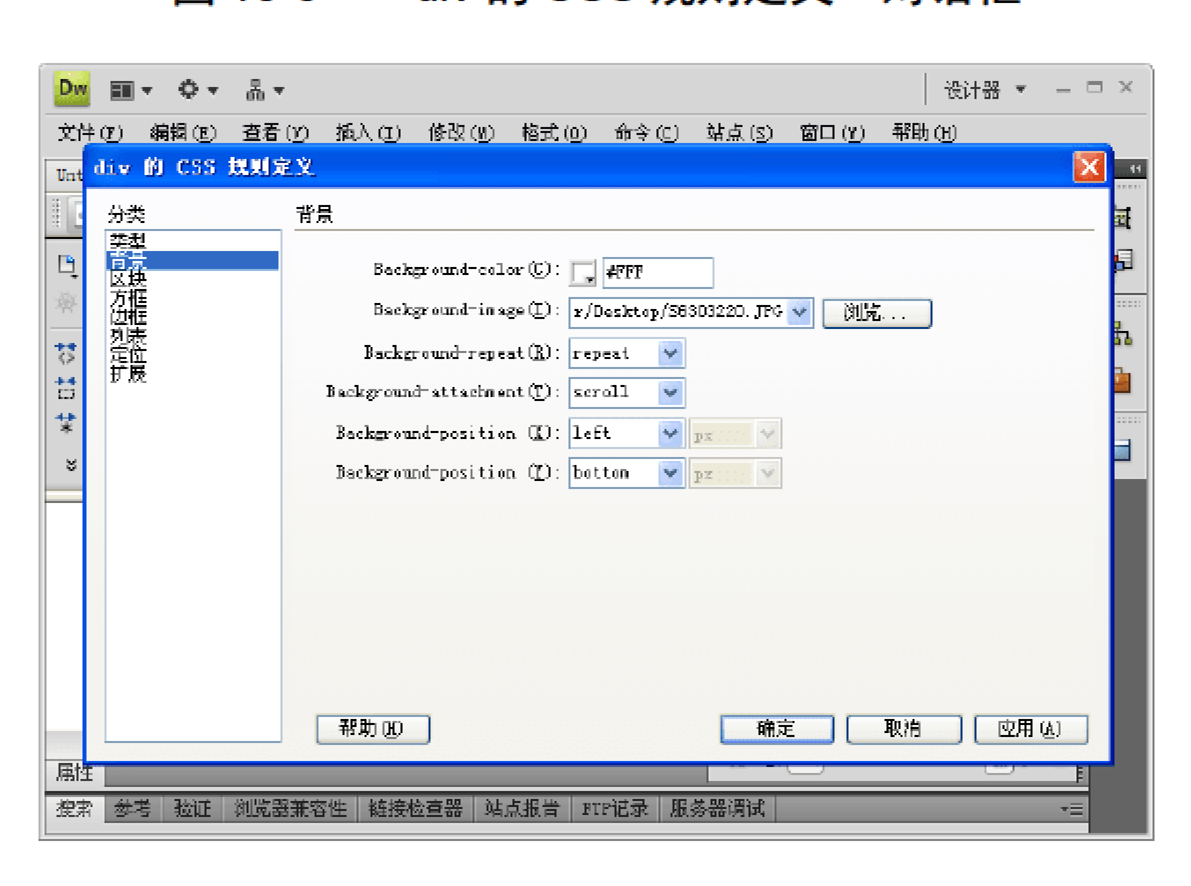


图 15-10　添加背景样式对话框

选择“分类”项后，在右边会出现对应的属性信息。例如，当我们选择了背景选项后，右边则会显示关于设置背景的所有属性。配置好相应的属性后单击“确定”按钮将会生成 CSS 样式信息，详细代码如下：

```
<html>
    <head>
<title>首页</title>
    <style type="text/css">
<!--
div {
}
body {
    background-attachment: fixed;
    background-color: #FFF;
    background-image: url(f474e2ef2d60f4d5b3fb9545.jpg);
    background-repeat: repeat;
    background-position: center center;
}
-->
    </style>
```

```
</head>
    <body>
        <div>正文内容</div>
    </body>
</html>
```

上述代码中的 background-attachment 属性、background-color 属性等，这些样式都是添加背景样式时使用 CSS 面板设置而生成的。

在 CSS 面板右下角还有 3 个功能按钮，即“附加样式表”、“编辑样式表”和“删除 CSS 规则”。这些按钮可以对页面的样式进行操作。例如，单击“删除 CSS 规则”按钮，可以通过 CSS 面板删除某个样式信息，如图 15-11 所示。

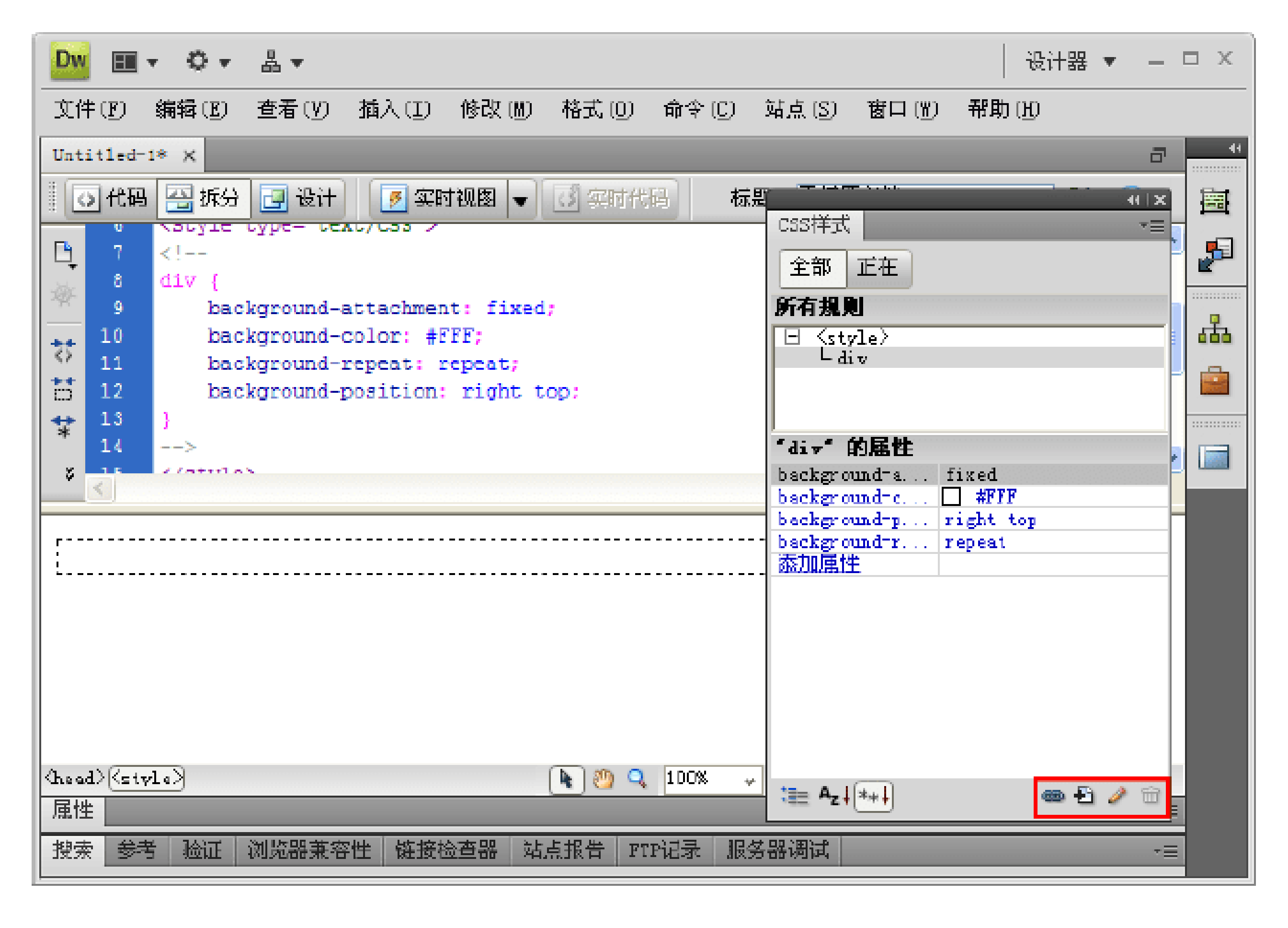


图 15-11　CSS 面板其他按钮

“附加样式表”按钮则可以将本地的 CSS 文件引入当前页面。使用该功能时需要单击此按钮打开如图 15-12 所示的对话框。

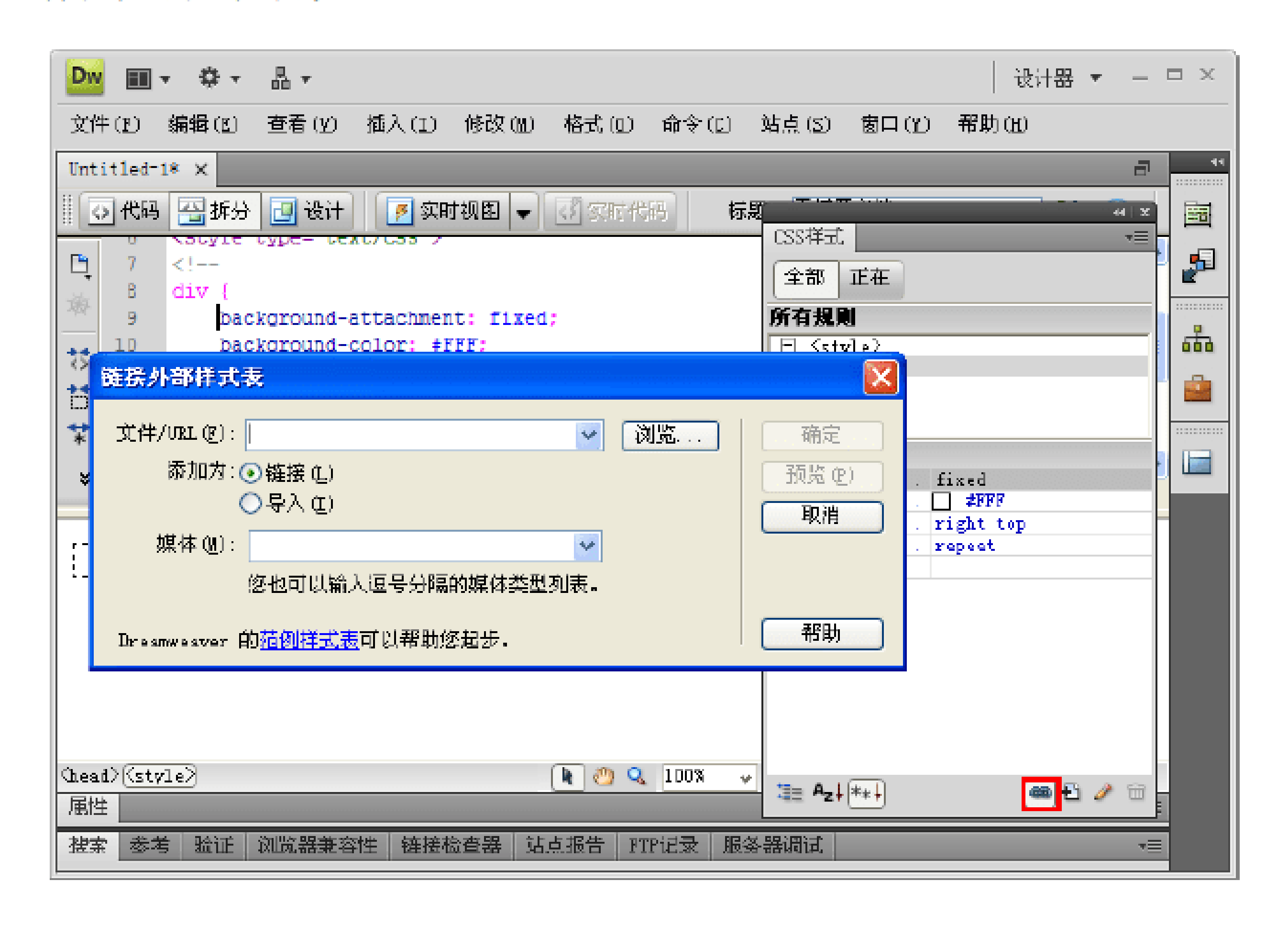


图 15-12　“连接外部样式表”对话框

在该对话框内找到需要引入的 CSS 样式文件，单击“确定”按钮即可完成引入操作。

15.1.3　实例描述

每个网站都会有这样的一个页面，用来为用户提供一些有用的信息。这些信息可以让用户在第一时间了解该网站为用户提供的服务，从而有选择地进行操作。那么，这个介绍页面必须设计得美观而且大方，可以通过使用 CSS 面板对此页面的介绍信息的字体进行设计。

15.1.4　实例应用

【例 15-1】网站介绍

在动手制作之前，首先需要将页面布局设计好。接下来通过使用 Dreamweaver 编辑器中的 CSS 面板工具添加样式。选择需要添加样式的标签，单击“新建 CSS 规则”按钮，打开“新建 CSS 规则”对话框，如图 15-13 所示。

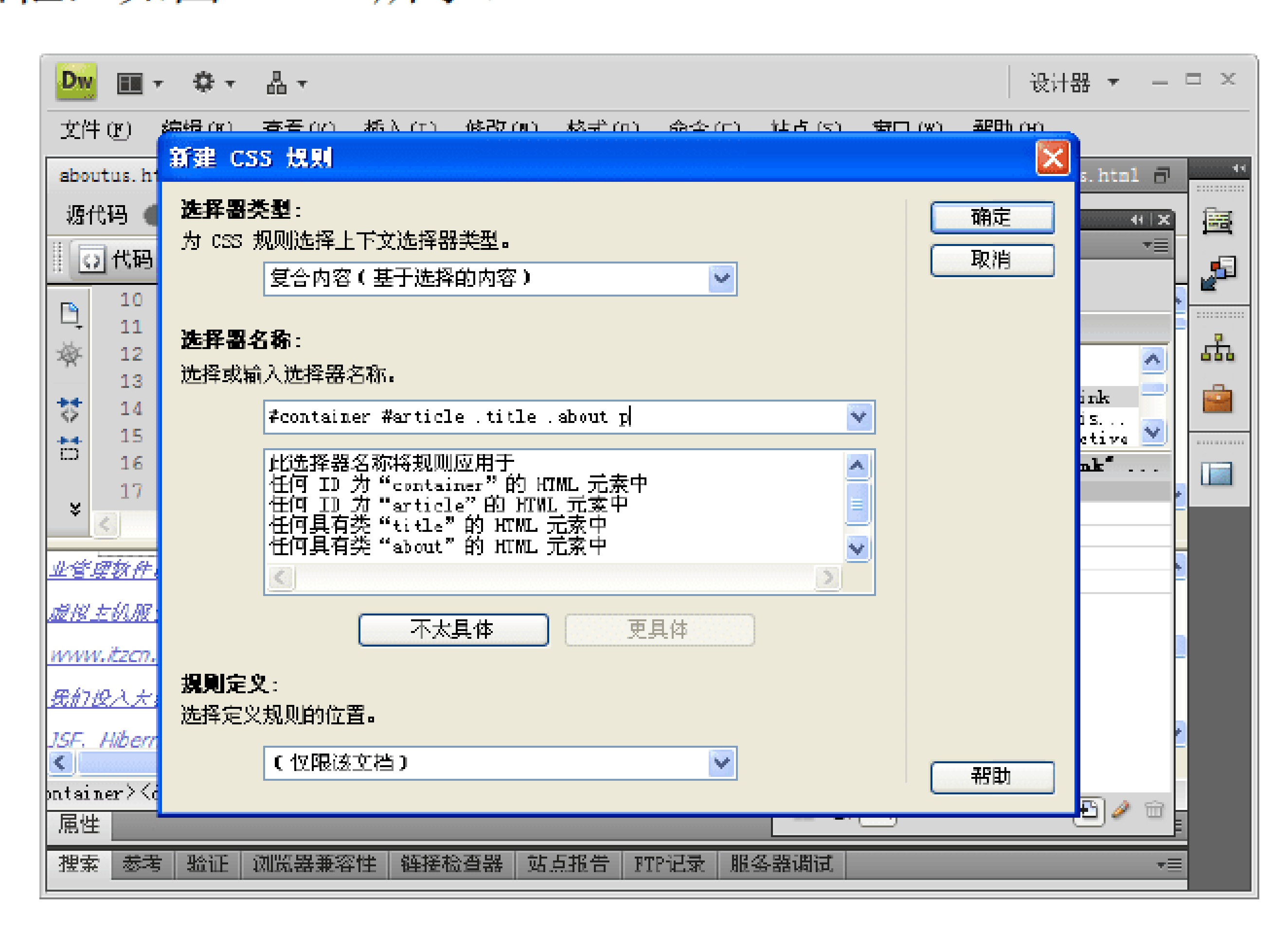


图 15-13　“新建 CSS 规则”对话框

此时在“新建 CSS 规则”对话框内会有“不太具体”和“更具体”两个按钮，用来设置 CSS 样式的范围。如果单击了“不太具体”按钮，则会以“选择添加样式的标签”为主添加样式，凡是使用该标签都会套用此样式。如果单击了“更具体”按钮，则会以 ID 为主显示样式范围。

接下来单击“确定”按钮进入“CSS 规则定义”对话框，选择需要的样式信息，如图 15-14 所示。

字体属性设置完成后，单击“确定”按钮，则会在页面最顶端的 style 标签内生成字体样式属性信息，详细代码如下所示：

```
<style type="text/css">
#container #article .title .about p {
    font-family: Tahoma, Geneva, sans-serif;/*字体类型*/
    font-size: 12px;/*字体大小*/
    font-style: italic;/*字体样式*/
```

```
    line-height: 25px;/*行高*/
    font-weight: normal;/*粗细*/
    font-variant: normal;/*变体*/
    text-transform: none;/*大小写*/
    color: #66C;/*字体颜色*/
    text-decoration: underline;/*修饰*/
}
</style>
```

上述代码中是通过设置字体样式而生成的样式代码，其中 font-size 用来设置字体的大小，font-weight 用来设置字体的粗细，color 用来设置字体的颜色等。

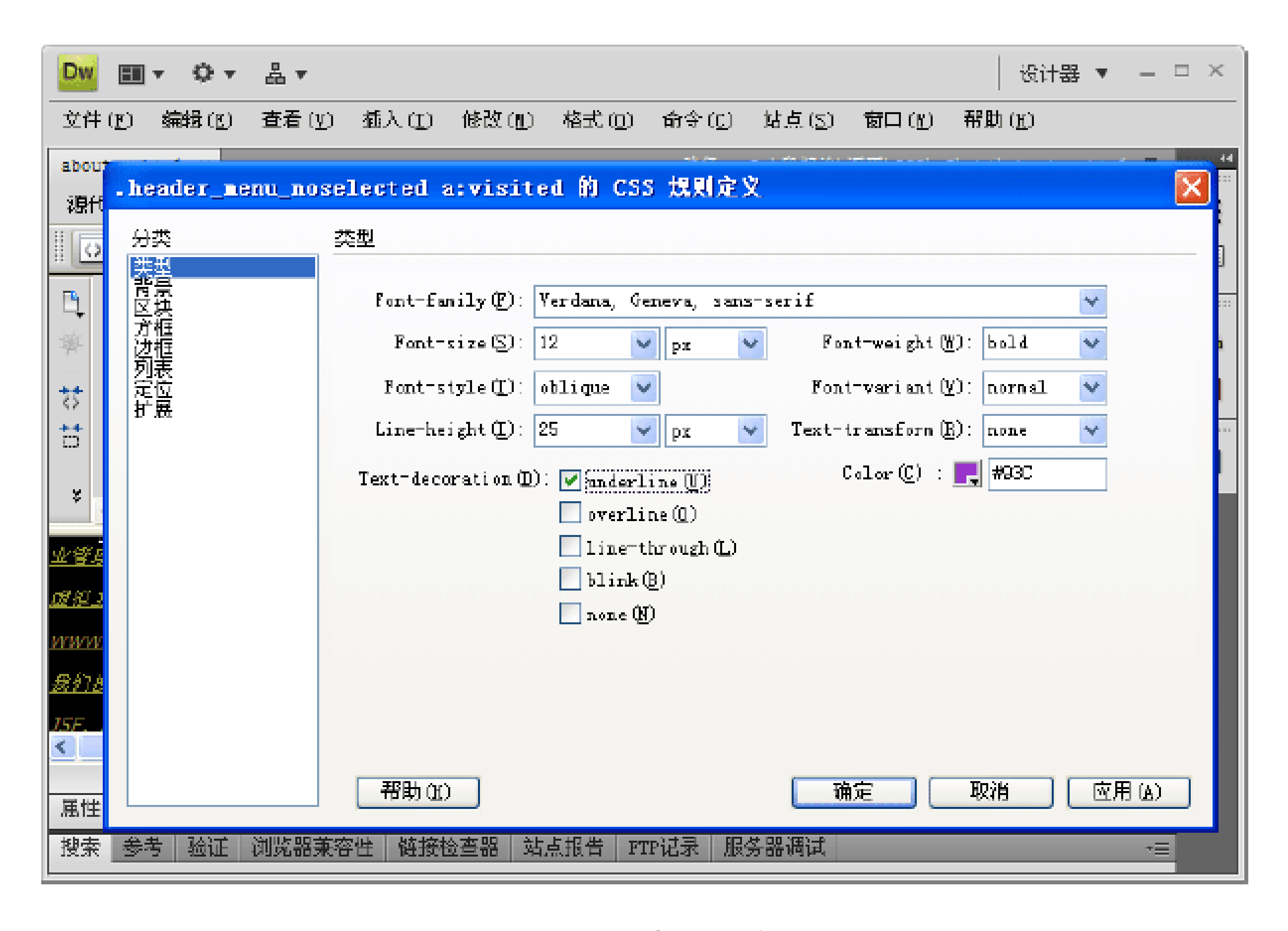


图 15-14 “CSS 规则定义”对话框

15.1.5 运行结果

在浏览器中运行 aboutus.html 页面，如图 15-15 所示。

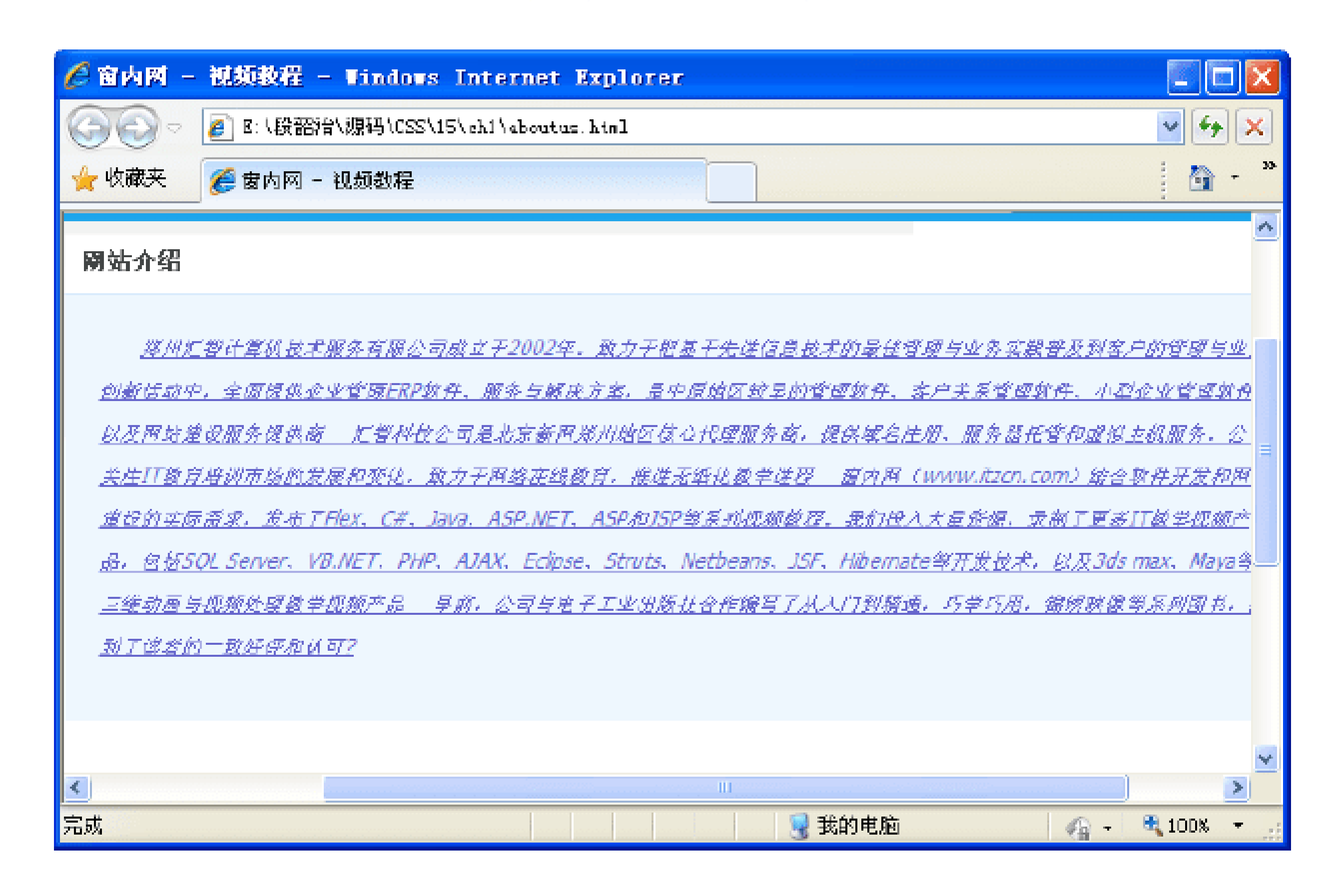


图 15-15 网站介绍运行结果

15.1.6　实例分析



源码解析：

上述实例中使用 CSS 面板创建了设置字体属性的一些信息。同样，还可以使用 CSS 面板来设置一些图片和页面的样式以及超链接等样式信息。因此大家在使用 CSS 面板创建样式时一定要活学活用。

15.2　将网站介绍页面样式保存至本地

这节讲解的内容非常简单，保存 CSS 文本到本地站点。现在仍然有很多人不知道将 CSS 保存到本地站点与页面内有什么区别，在本节就来讲解将 CSS 代码分离成单独文件有什么好处。



视频教学：光盘/视频/15/15.2　将网站介绍页面样式保存至本地.avi　　长度：5 分钟

15.2.1　基础知识——保存 CSS 文件到本地站点

之所以将 CSS 文件单独提取出来作为一个文件保存在本地站点，是因为这样可以大大地减少页面上的代码量，以方便以后对页面进行维护。有时，可能页面布局比较复杂，使用样式比较多，如果把所有的样式信息都写入页面，那么会加大后期维护的难度；如果将整个站点的样式信息都保存到一个 CSS 文件中，当页面布局出现问题或者需要扩展布局样式时，只需要打开 CSS 样式文件修改即可。接下来就讲解如何使用 Dreamweaver 工具将书写好的 CSS 样式以文件的形式保存到本地站点。

首先在页面内新建 CSS 规则，如图 15-16 所示。



图 15-16　新建 CSS 规则

单击“新建 CSS 规则”按钮打开“新建 CSS 规则”对话框，如图 15-17 所示。

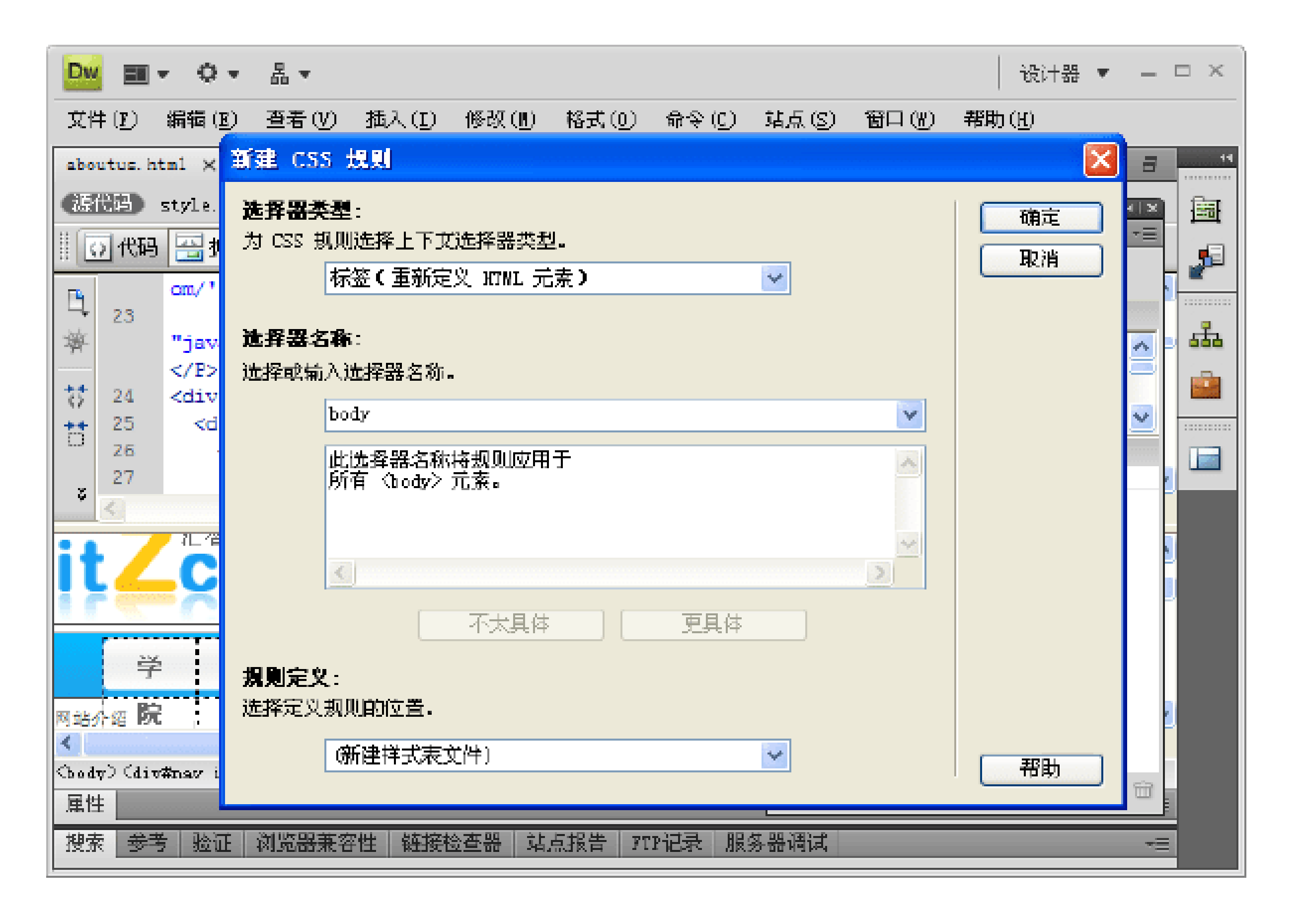


图 15-17 “新建 CSS 规则”对话框

在“规则定义”下拉列表框内默认有两个值，即“仅限该文档”和“新建样式表文件”选项。当页面内已经嵌入了一个外部的 CSS 文件时，在这里面则会显示嵌入外部样式文件的名称选项。在这里启用“新建样式表文件”选项，单击“确定”按钮打开“将样式表文件另存为”对话框，如图 15-18 所示。

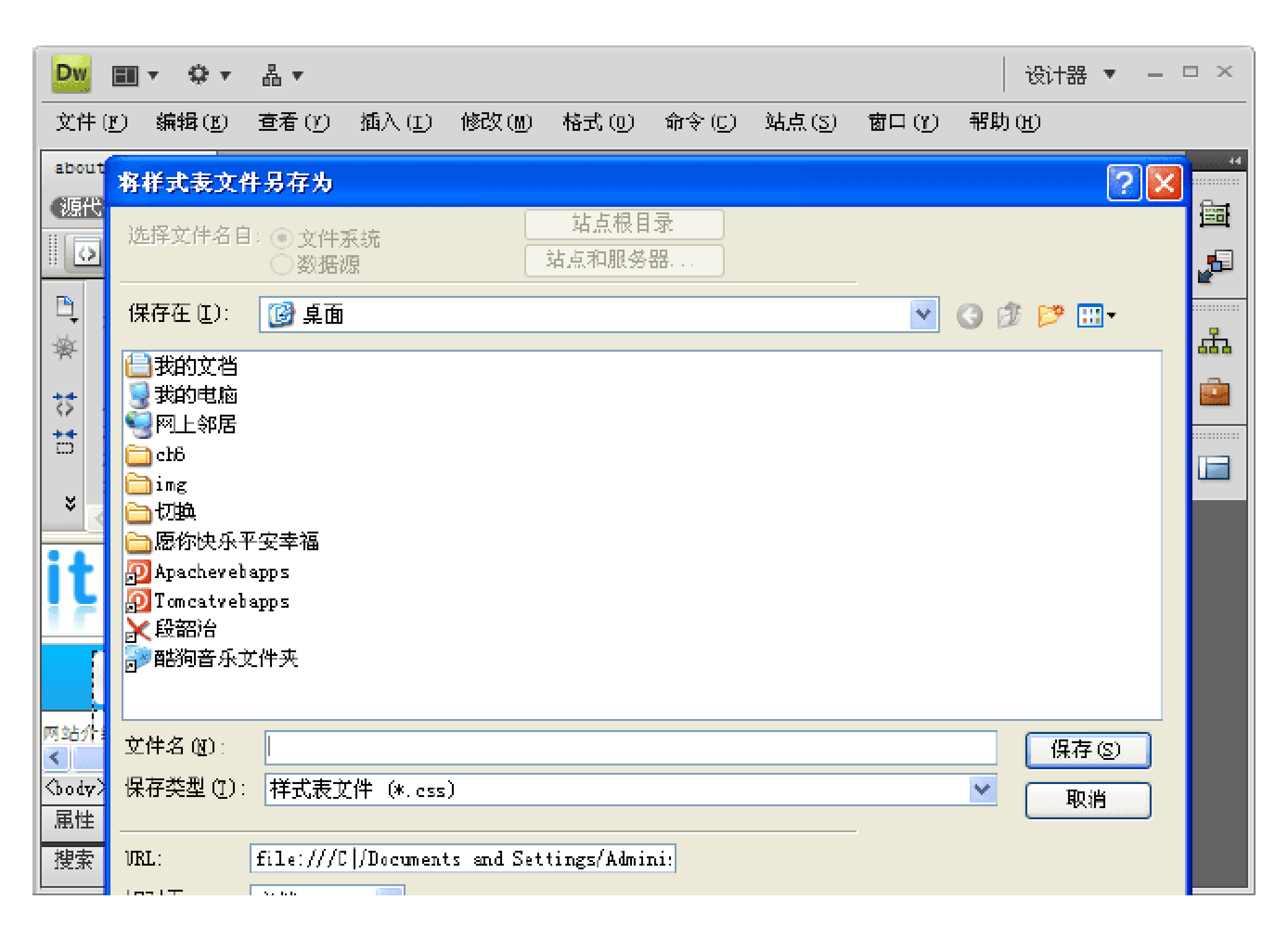


图 15-18 “将样式表文件另存为”对话框

在该对话框中选择指定保存路径和文件名等信息，单击“保存”按钮，打开“CSS 规则定义”对话框，如图 15-19 所示。

在该对话框中保持默认参数不变，单击“确定”按钮，将所有样式信息以文件的形式保存到 style.css 文件中，style.css 文件详细代码如下：

```
@charset "utf-8";
body {
    font-family: Verdana, Geneva, sans-serif;
    font-size: 12px;
```

```
    font-style: normal;
    line-height: 20px;
    font-weight: normal;
    font-variant: normal;
    text-transform: none;
    color: #C3F;
    text-decoration: underline;
}
```

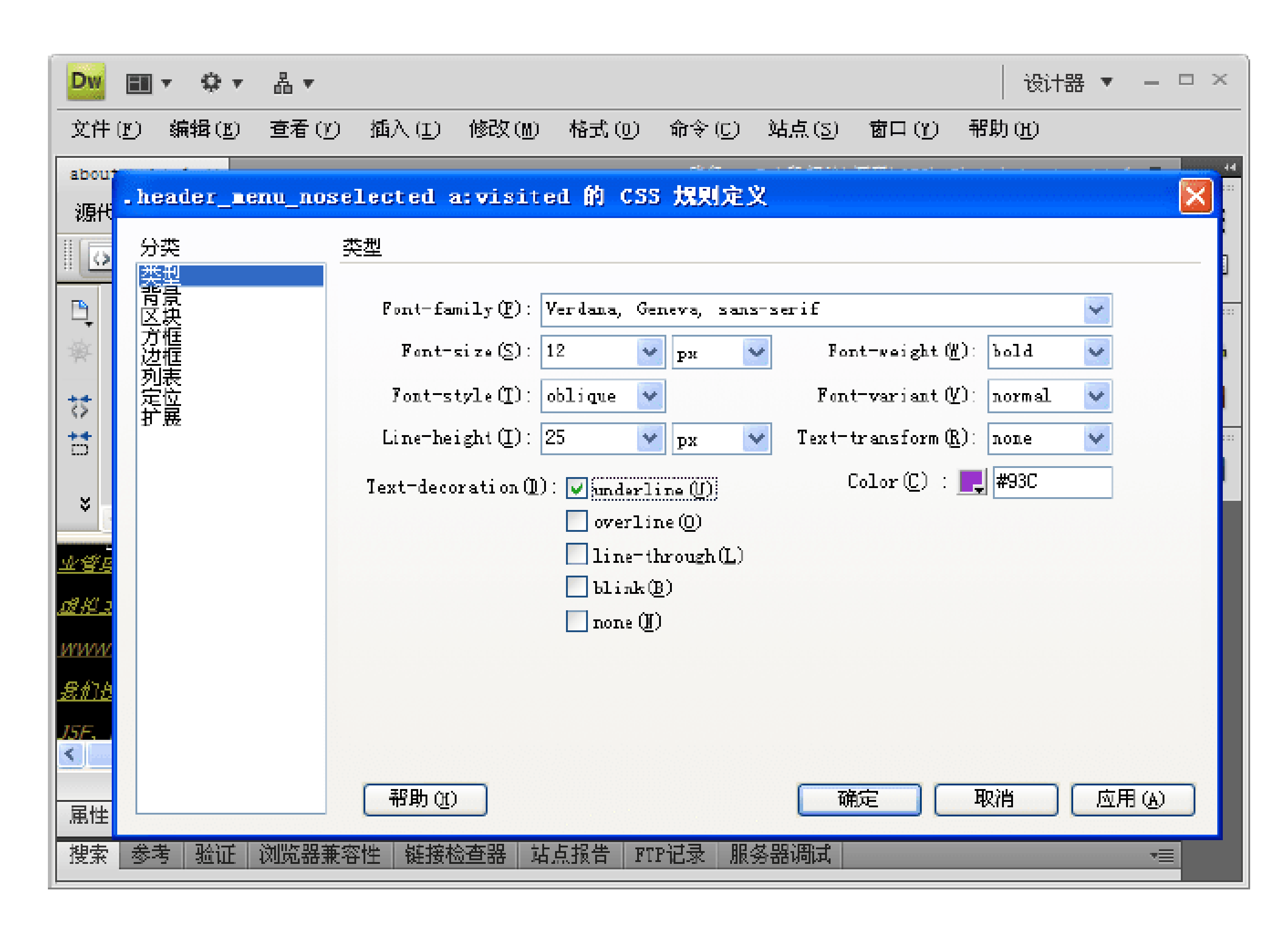


图 15-19　“CSS 规则定义”对话框

上述代码保存在 style.css 样式表的样式信息中，在页面内会自动生成以下代码：

```
<link href="style.css" rel="stylesheet" type="text/css" />
```

该代码大家肯定不陌生，因为它是用来将外部样式表引入当前页面的。

15.2.2　实例描述

在 15.1 节讲解了如何设计网站介绍页面，本节仍然以“网站介绍页面”为例进行讲解。在本实例中，将要把生成的 CSS 文件保存到本地服务器，同时页面也将引用该文件样式信息。此外，通过该样式文件中的样式信息将会映射页面的布局和显示方式，从而完成页面代码的合理分类。

15.2.3　实例应用

【例 15-2】网站介绍页面

首先，通过 Dreamweaver 工具打开窗内网网站介绍页面。然后打开“CSS 样式”面板新建样式规则，如图 15-20 所示。

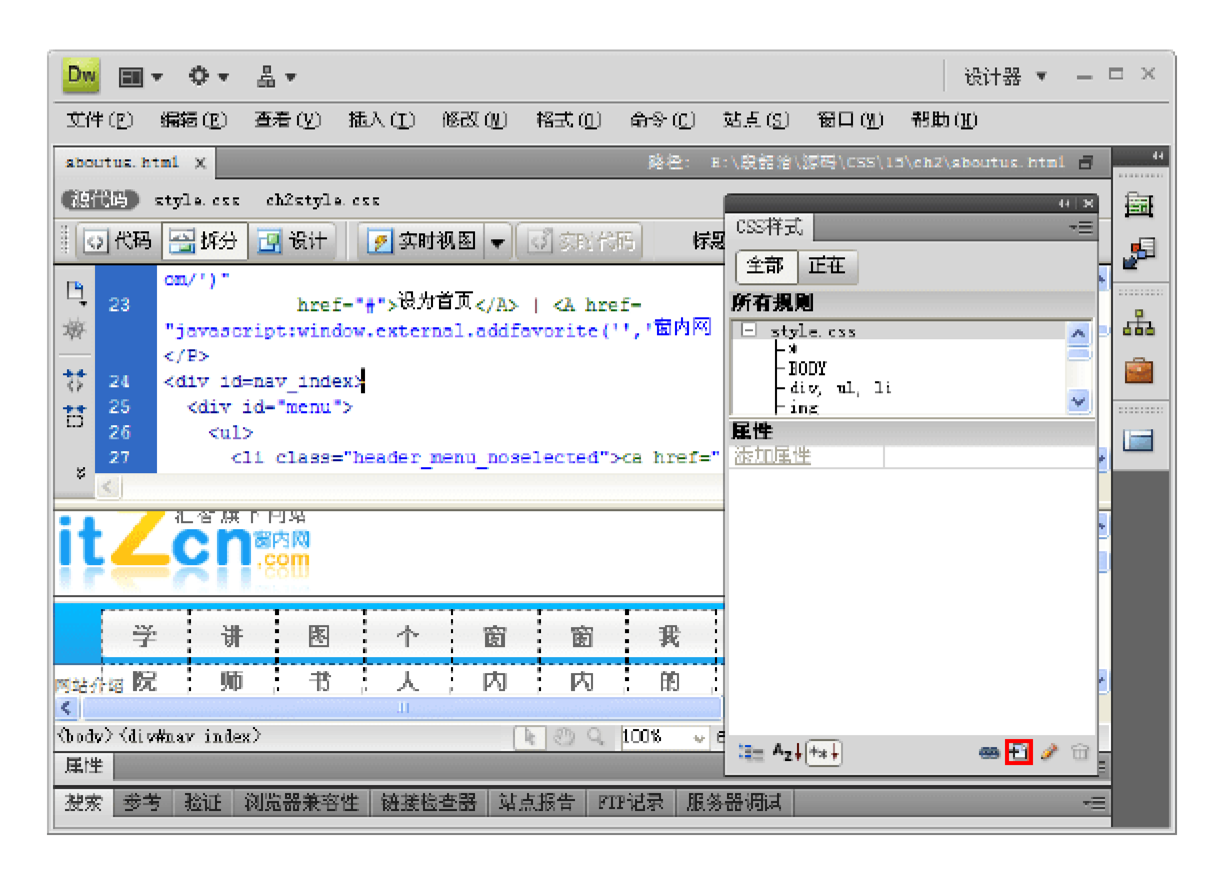


图 15-20　CSS 样式面板

单击“新建 CSS 规则”按钮打开 “新建 CSS 规则”对话框，如图 15-21 所示。

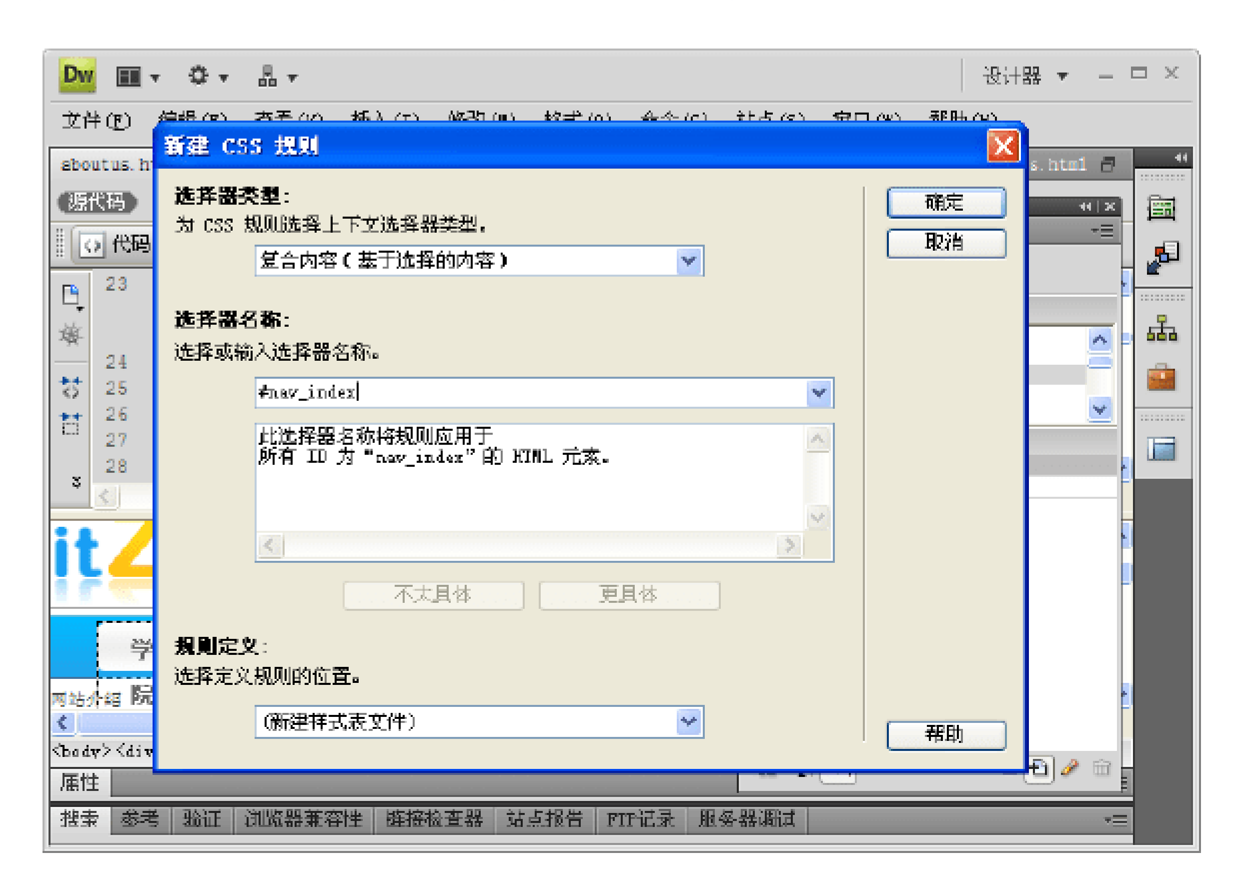


图 15-21　“新建 CSS 规则”对话框

下面设置一些基本的规则信息。以 p 元素为主样式，将样式信息以新建样式表文件保存至本地，单击“确定”按钮，打开“将样式表文件另存为”对话框，如图 15-22 所示。

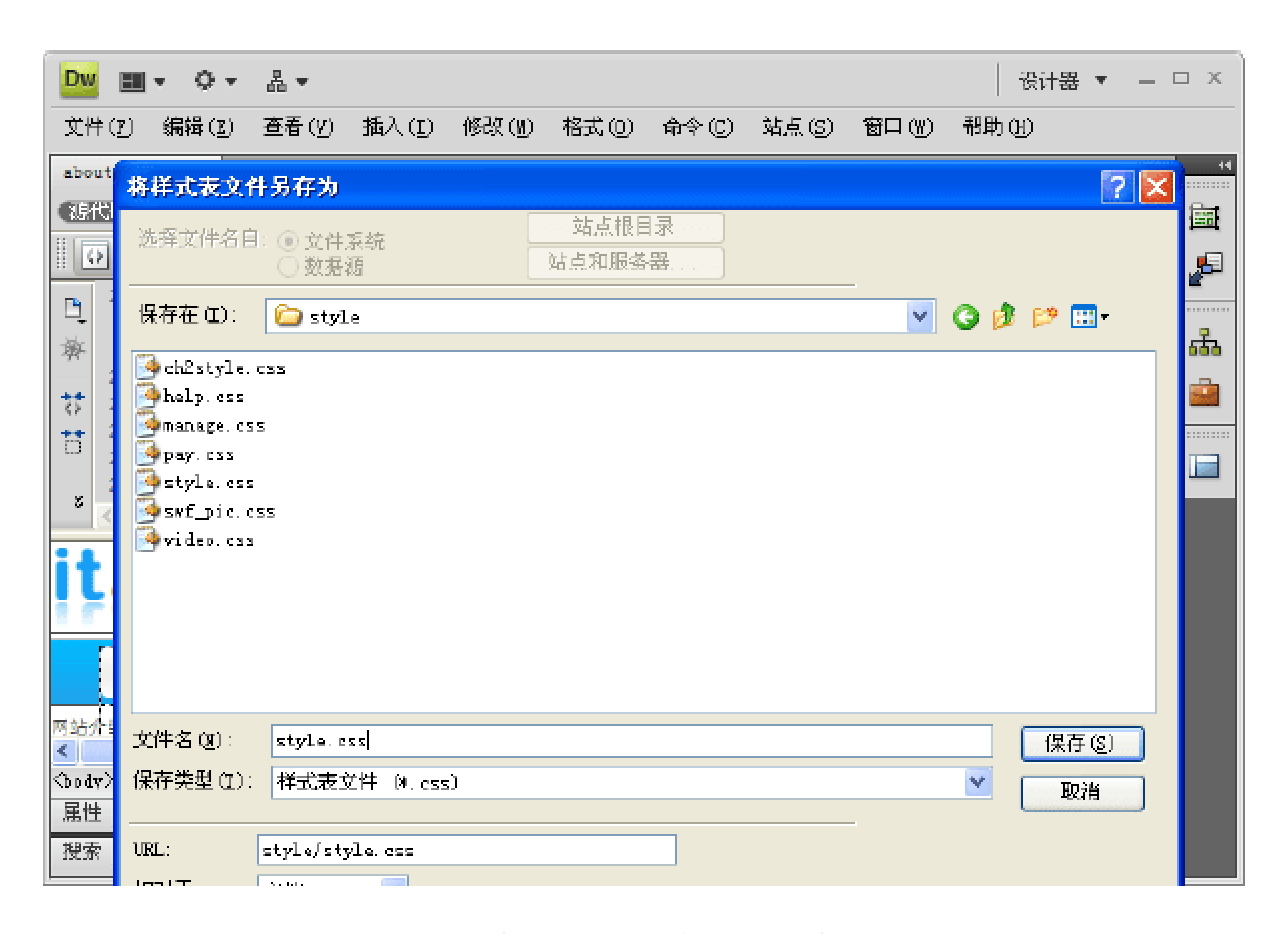


图 15-22　“将样式表文件另存为”对话框

将样式文件保存至本地项目下的 style 下，将文件命名为 ch2style.css，单击“保存”按钮进行保存。此时将会弹出“CSS 规则定义(ch2style.css 中)”对话框，如图 15-23 所示。

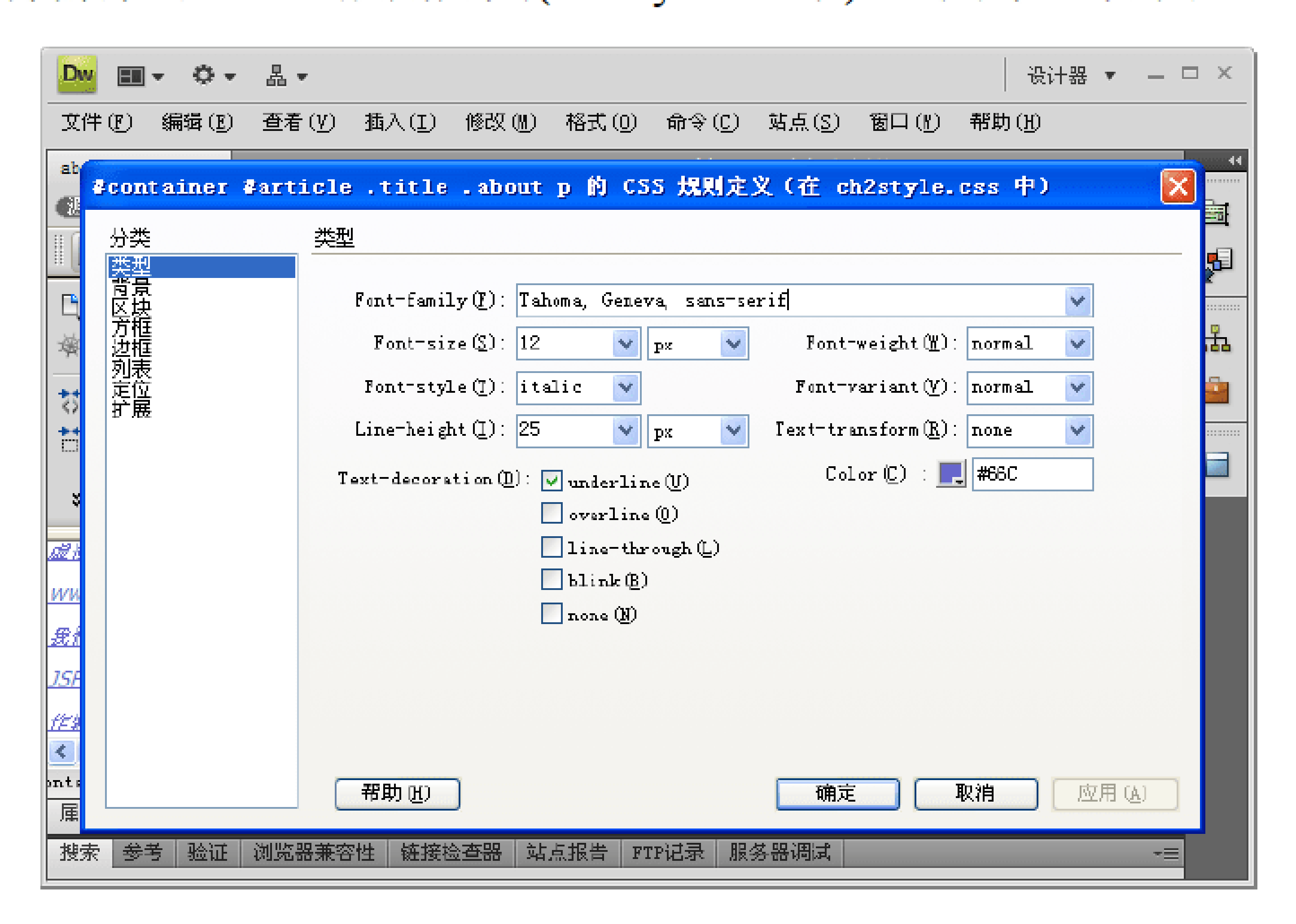


图 15-23 “CSS 规则定义(ch2style.css 中)”对话框

设置相关的样式信息，单击“确定”按钮，在项目 style 目录下将会生成名为 ch2style.css 的文件，该文件能保存的代码如下：

```
@charset "utf-8";
#container #article .title .about p {
    font-family: Tahoma, Geneva, sans-serif;/*字体类型*/
    font-size: 12px;/*字体大小*/
    font-style: italic;/*字体样式*/
    line-height: 25px;/*行高*/
    font-weight: normal;/*粗细*/
    font-variant: normal;/*变体*/
    text-transform: none;/*大小写*/
    color: #66C;/*字体颜色*/
    text-decoration: underline;/*修饰*/
}
```

除了上述代码外，同时也会生成如下代码：

```
<link href="style/ch2style.css" rel="stylesheet" type="text/css" />
```

15.2.4 运行结果

接下来保存页面，在浏览器中运行 aboutus.html 页面，效果如图 15-24 所示。

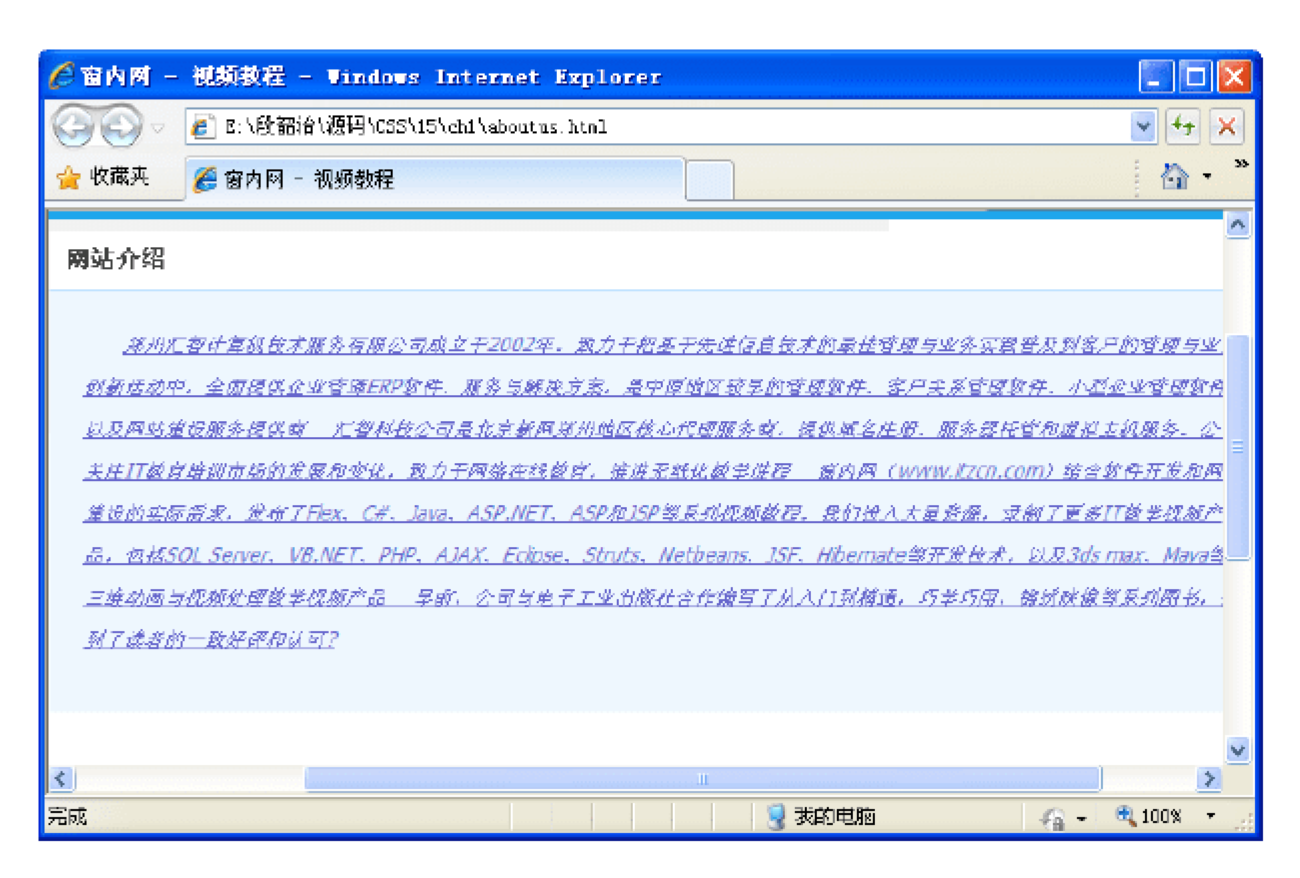


图 15-24　网站介绍运行效果

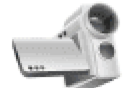
15.2.5　实例分析


源码解析：

在这个实例中通过设置页面字体样式生成 CSS 代码，将代码以文件的形式保存至本地服务器，以便于以后维护，因此该项目通过使用 Dreamweaver 工具中的 CSS 面板生成 CSS 文件。

15.3　为网站介绍页面添加背景颜色

在网站开发的过程中，难免要对一个已有的样式进行修改。为了方便程序员把时间充分放在程序的设计上，CSS 样式面板提供了在原有样式上直接添加、修改和删除等操作。本节将会讲解如何直接为网站页面添加背景颜色。


视频教学：光盘/视频/15/15.3　为网站介绍页面添加背景颜色.avi　　　长度：4 分钟

15.3.1　实例描述

本实例主要通过在面板中直接添加 CSS 背景属性，生成 CSS 样式文件来制作，同样，还借助 15.2 节中的实例进行整个项目的设计。另外要知道的是，所谓的直接添加 CSS 背景属性就是将原有的属性文件选中，通过 CSS 面板直接添加背景样式，而不是重新创建样式表。

15.3.2　实例应用

【例 5-3】为网站介绍页面添加背景颜色

首先通过 Dreamweaver 打开之前制作的介绍页面，再在页面添加 CSS 背景属性，打开

CSS 样式面板，在该面板内有“所有规则”和“属性”两个模块，如图 15-25 所示。

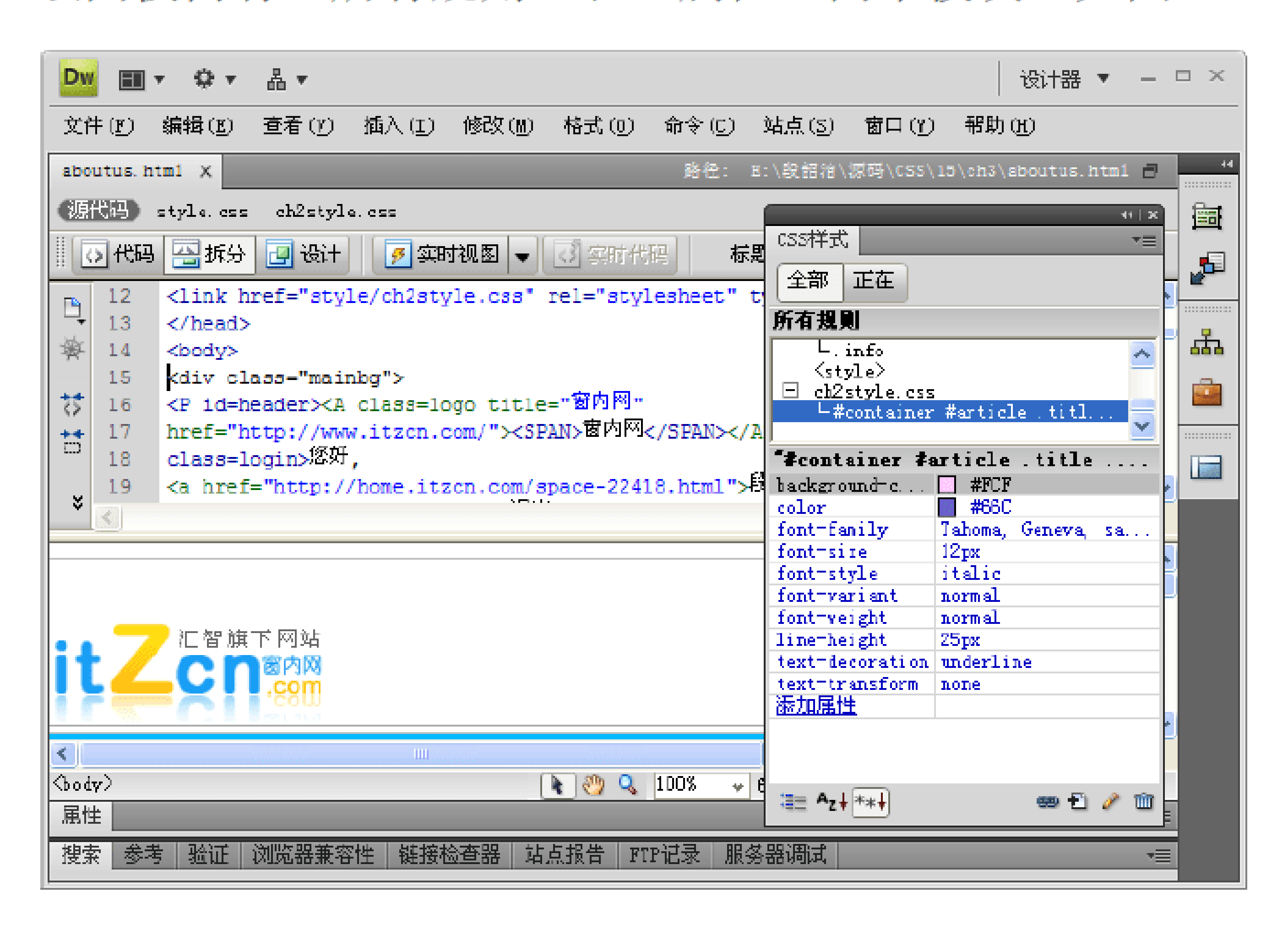


图 15-25　CSS 样式面板

“所有规则”模块用于在显示之前创建样式名称，当选择定义好的 CSS 样式名称时，该属性模块则会显示相关属性。此时在所有属性的最下端有一个“添加属性”的超链接，单击超链接就可以对该样式直接添加样式属性，如图 15-26 所示。

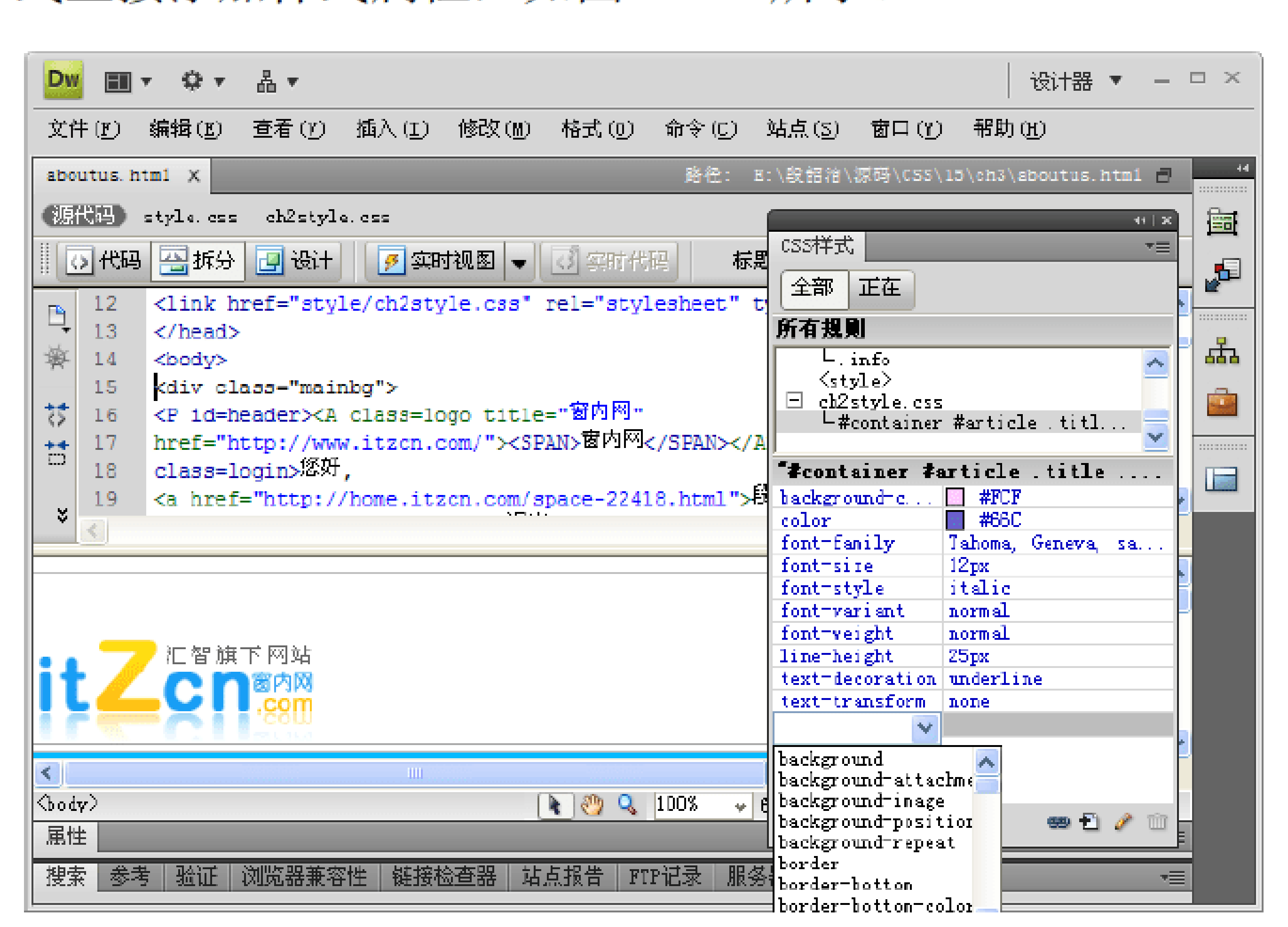


图 15-26　直接添加背景属性

在图片中设置该样式的 background-color 属性，从而来改变页面的背景颜色。使用同样的方法还可以设置页面的背景图片等相关样式属性，此时将会在定义好的 CSS 文件内添加如下代码：

```
background-color: #DAF5FE;
```

如果在编辑过程中发现选错了属性，还可以进行删除和修改，如图 15-27 所示。

在这里只需在需要进行删除或编辑的样式中进行操作即可，经过修改或者删除操作后的 CSS 文件也会做出相应的更改。

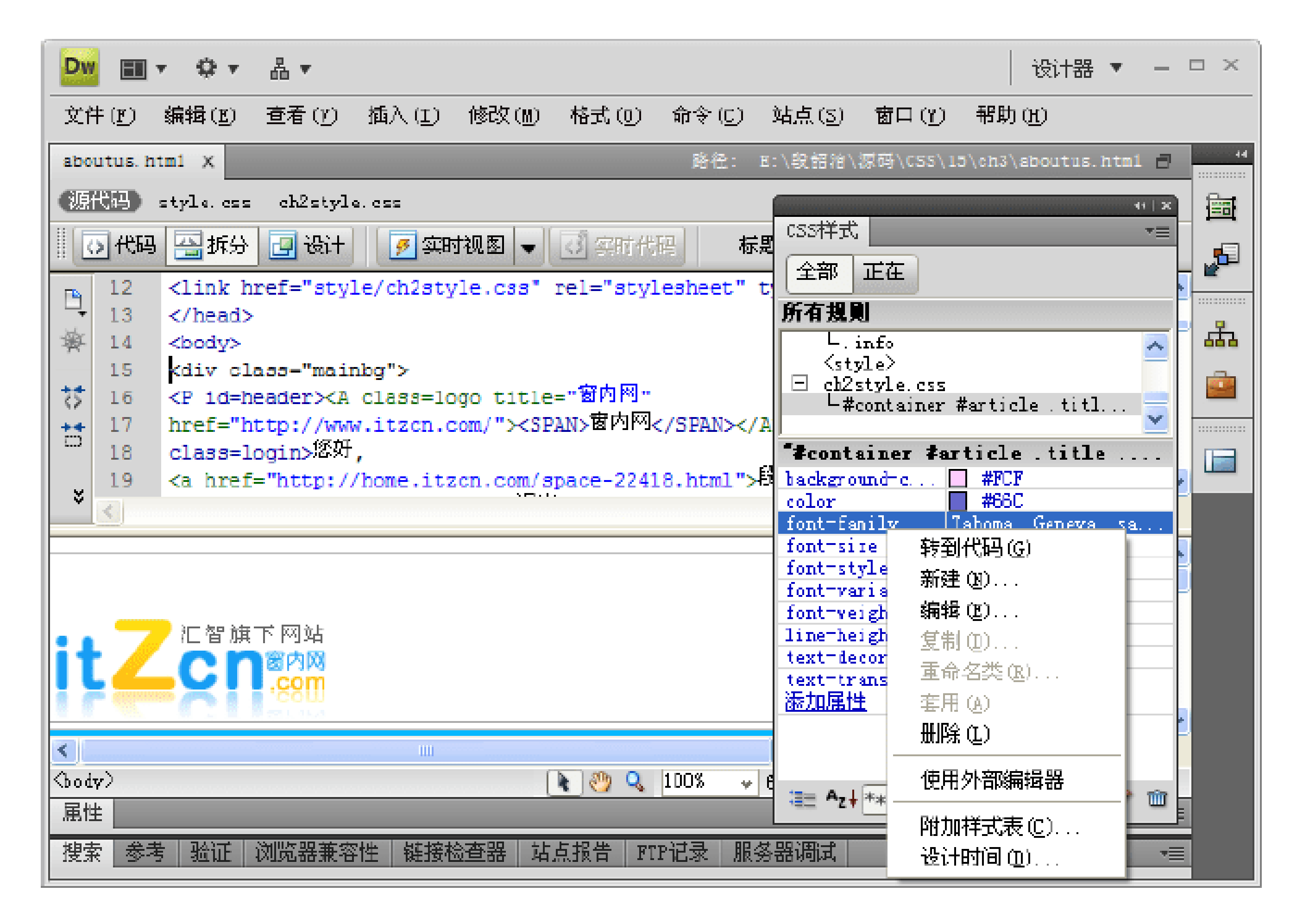


图 15-27　编辑和删除功能

15.3.3　运行结果

接下来保存修改后的结果，在浏览器中运行 aboutus.html 页面，效果如图 15-28 所示。

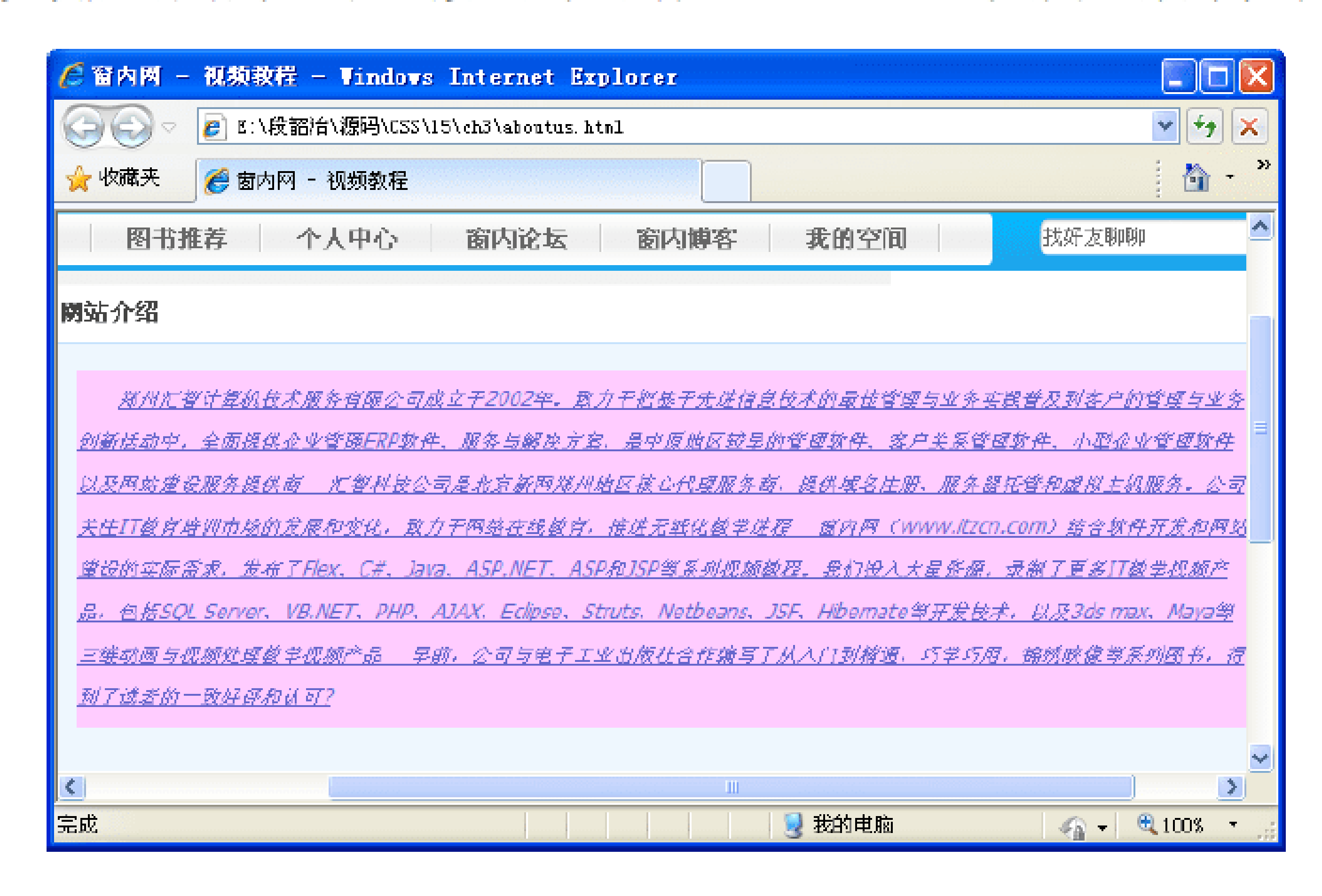


图 15-28　网站介绍页面

15.3.4　实例分析



源码解析：

在这个实例中，通过定义好的样式，可以直接为页面添加该样式的背景属性和相关的样式信息。同样在使用的过程中，如果发现不需要的属性或者需要修改的属性，都可以直接通过 CSS 面板进行删除和修改。这样就避免了再次打开文件修改页面的样式信息的过程。

15.4　设置首页导航

菜单栏组件是一组可导航的菜单按钮，当访问者将鼠标指向其中的某个按钮上时，将显示相应的子菜单。使用菜单栏可以在紧凑的空间中显示大量可导航信息，并使访问者无须深入浏览即可了解网站中提供的内容。

视频教学： 光盘/视频/15/15.4　设置首页导航.avi　　长度：8 分钟

Spry 框架是一个可以用来构建更加丰富的 Web 页的 JavaScript 和 CSS 库。使用该框架，可以显示 XML 数据，并可创建用来显示动态数据的交互式页面元素，而无需刷新整个页面。以前创建 Spry 框架是通过代码完成的，这需要有一定的 JavaScript 基础知识的用户使用。而 Dreamweaver CS4 在设计视图中新增了创建 Spry 框架的组件，使复杂的 Spry 框架创建变成了简单的可视化操作。

Spry 框架支持一组用标准 HTML、CSS 和 JavaScript 编写的可重用组件。Spry 组件是一个页面元素，通过用户交互来提供更加丰富的用户体验。每一个 Spry 组件均由以下几个部分组成。

- 组件结构：用来定义组件结构组成的 HTML 代码块。
- 组件行为：用来控制组件如何响应用户启动事件的 JavaScript。
- 组件样式：用来指定组件外观的 CSS。

在 Dreamweaver CS4 中，所有的 Spry 组件集中在“插入”工具栏的 Spry 选项卡中，方便归类与创建，如图 15-29 所示。

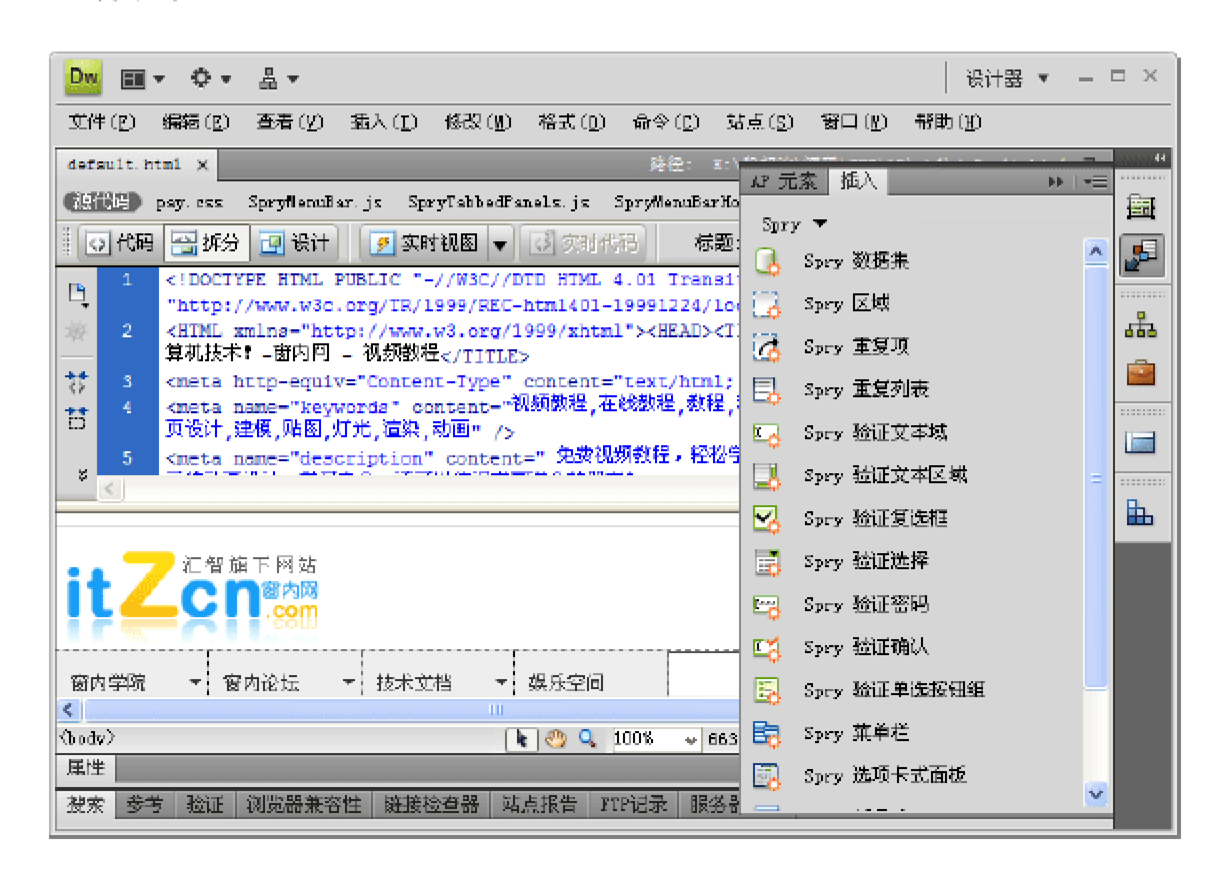


图 15-29　“插入”工具栏中的 Spry 组件

Spry 框架中的每个组件都与唯一的 CSS 和 JavaScript 文件相关联。CSS 文件中包含设置组件样式所需的全部信息，而 JavaScript 文件则赋予组件功能。当使用 Dreamweaver 界面插入组件时，Dreamweaver 会自动将这些文件链接到页面，以便组件中包含该页面的功能和样式。

与给定组件相关联的 CSS 和 JavaScript 文件根据该组件命名。因此，用户很容易判断哪些文件对应于哪些组件。

提示：当用户在已保存的页面中插入组件时，Dreamweaver 会在站点中创建一个 SpryAssets 目录，并将相应的 JavaScript 和 CSS 文件保存到其中。

15.4.1 基础知识——插入菜单栏

Dreamweaver CS4 允许用户插入两种类型的菜单栏：垂直和水平，这两者插入方法相同。方法是：单击“布局”选项卡中的“Spry 菜单栏”按钮 Spry 菜单栏，然后在“Spry 菜单栏”对话框中通过“水平”或者“垂直”单选按钮选择不同的布局选项，如图 15-30 和图 15-31 所示。

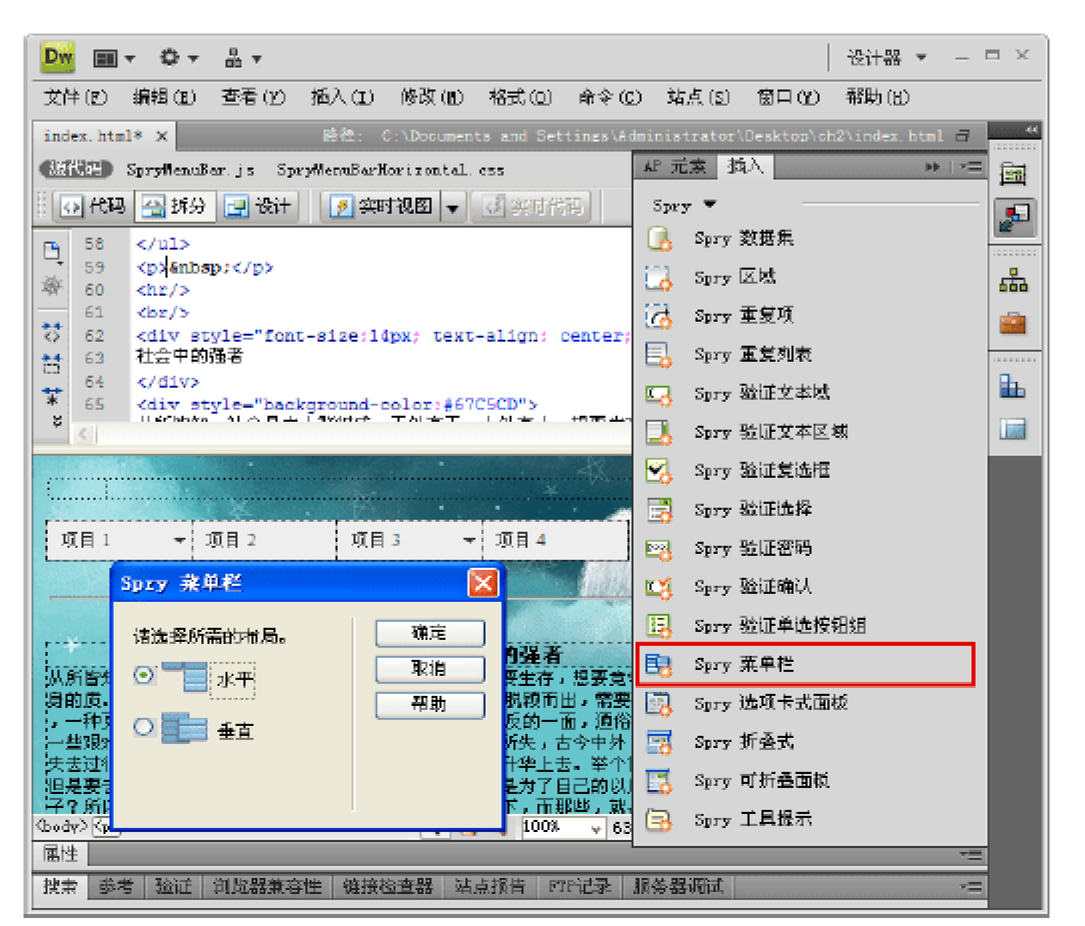


图 15-30 水平 Spry 菜单

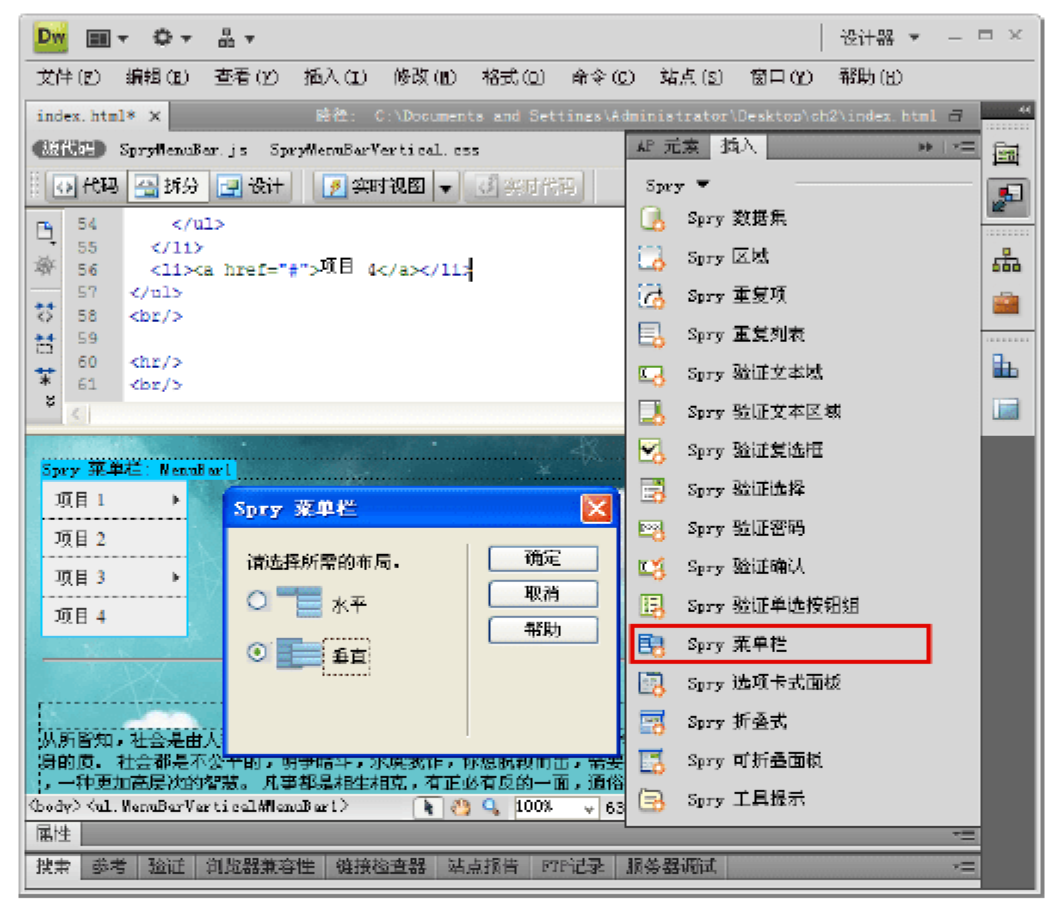


图 15-31 垂直 Spry 菜单

保存文档后，就可以预览默认样式的水平或者垂直布局的菜单栏效果，如图 15-32 和图 15-33 所示。当某个菜单栏目右侧出现小三角时，说明该栏目还包括子菜单。



图 15-32 垂直 Spry 菜单运行结果



图 15-33 水平 Spry 菜单运行结果

菜单栏组件的 HTML 中包含一个外部 ul 标签。该标签中对于每个顶级菜单项都包含一个 li 标签，而顶级菜单项(li)又包含用来为每个菜单项定义子菜单的 ul 和 li 标签。另外子菜单中

同样可以包含子菜单。顶级菜单和子菜单可以包含任意多个子菜单项。

15.4.2　基础知识——编辑菜单栏

创建 Spry 菜单栏组件后，单击 Spry 菜单栏，在“属性”面板中显示菜单条属性，如图 15-34 所示。

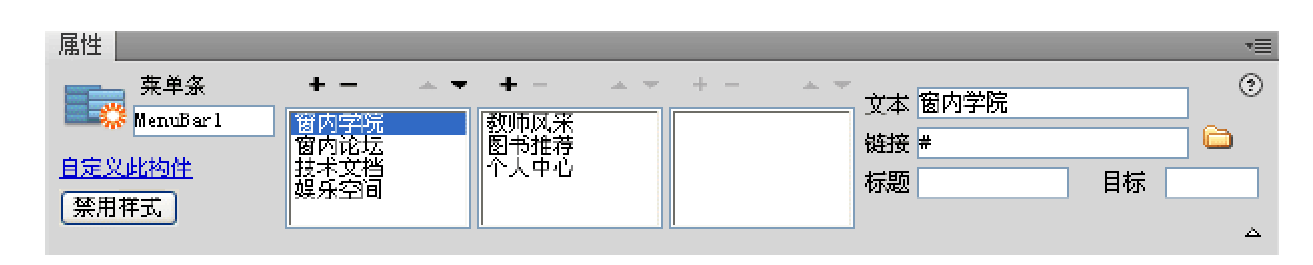


图 15-34　Spry 菜单栏“属性”面板

其中，Spry 菜单栏“属性”面板中的各个选项及作用如下。

- 菜单栏名称：默认菜单栏名称为 MenuBar1，该名称不能以汉字命名，可以使用字母或者数字。
- 禁用样式：单击该按钮，菜单栏变成项目列表，并且按钮名称更改为“启用样式”。
- 菜单栏目：在“属性”面板中包括主菜单栏目、一级菜单栏目与二级菜单栏目。
- 文本：在该选项文本框中设置栏目名称。
- 链接：该选项可以为菜单栏目添加链接文件，默认情况下为空链接。单击“浏览”按钮 可以选择链接文件。
- 目标：该选项指定要在何处打开所链接的页面。
- 标题：该选项为菜单栏目提示文本。

1. 调整菜单栏目

选中 Spry 菜单栏，“属性”面板中显示的是默认菜单栏目，并且选中的是主菜单第一个栏目名称。在主菜单栏目区域中单击向下箭头一次，可以将该栏目与第二个栏目交换位置，而相应的子菜单栏目也会随之变位，如图 15-35 所示。

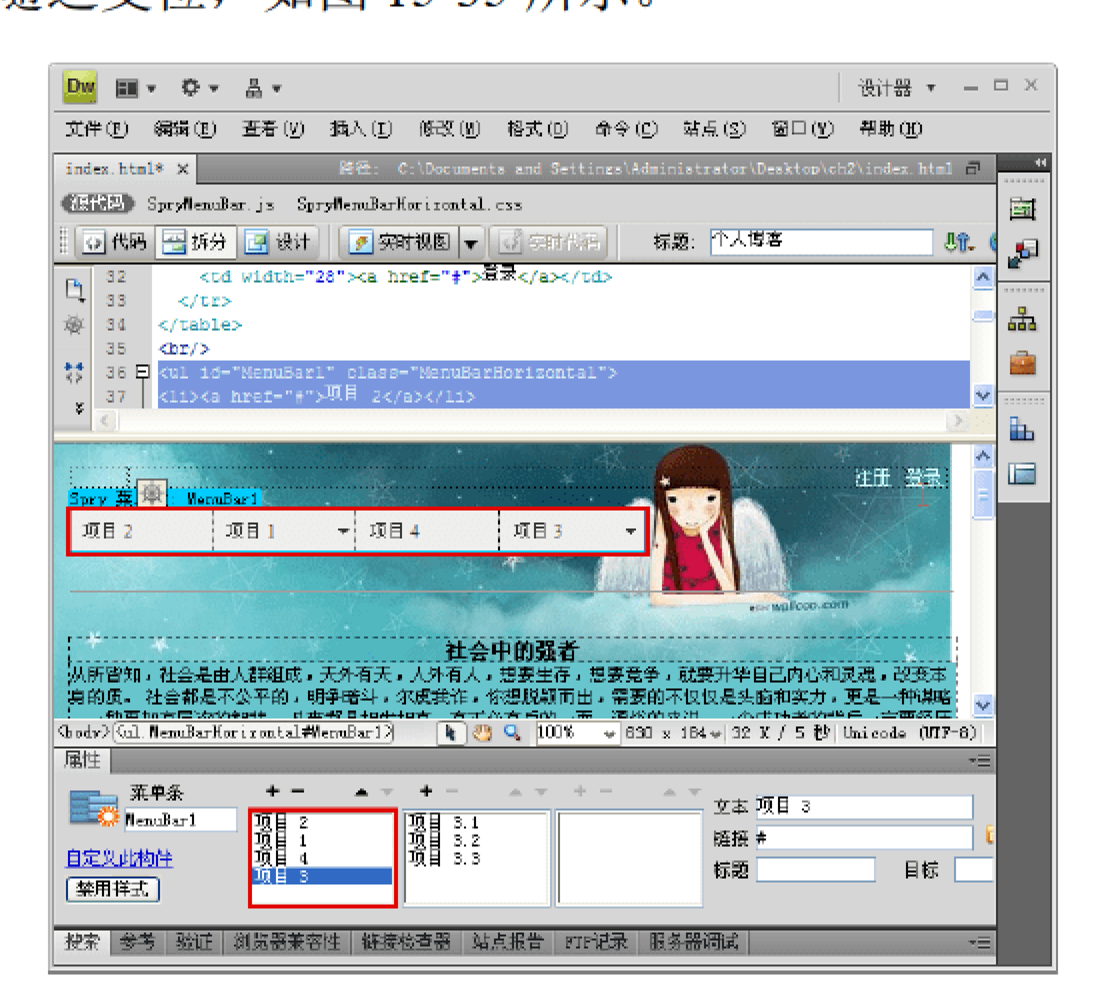


图 15-35　更改菜单项的顺序



在一级菜单栏目与二级菜单栏目中同样可以更改菜单项顺序，并且相应的子菜单随之变位。

在默认情况下，主菜单包括4个栏目。选中其中一个菜单栏目名称，单击“添加菜单项”按钮+，在其下方添加“无标题项目”的菜单栏目，如图15-36所示。

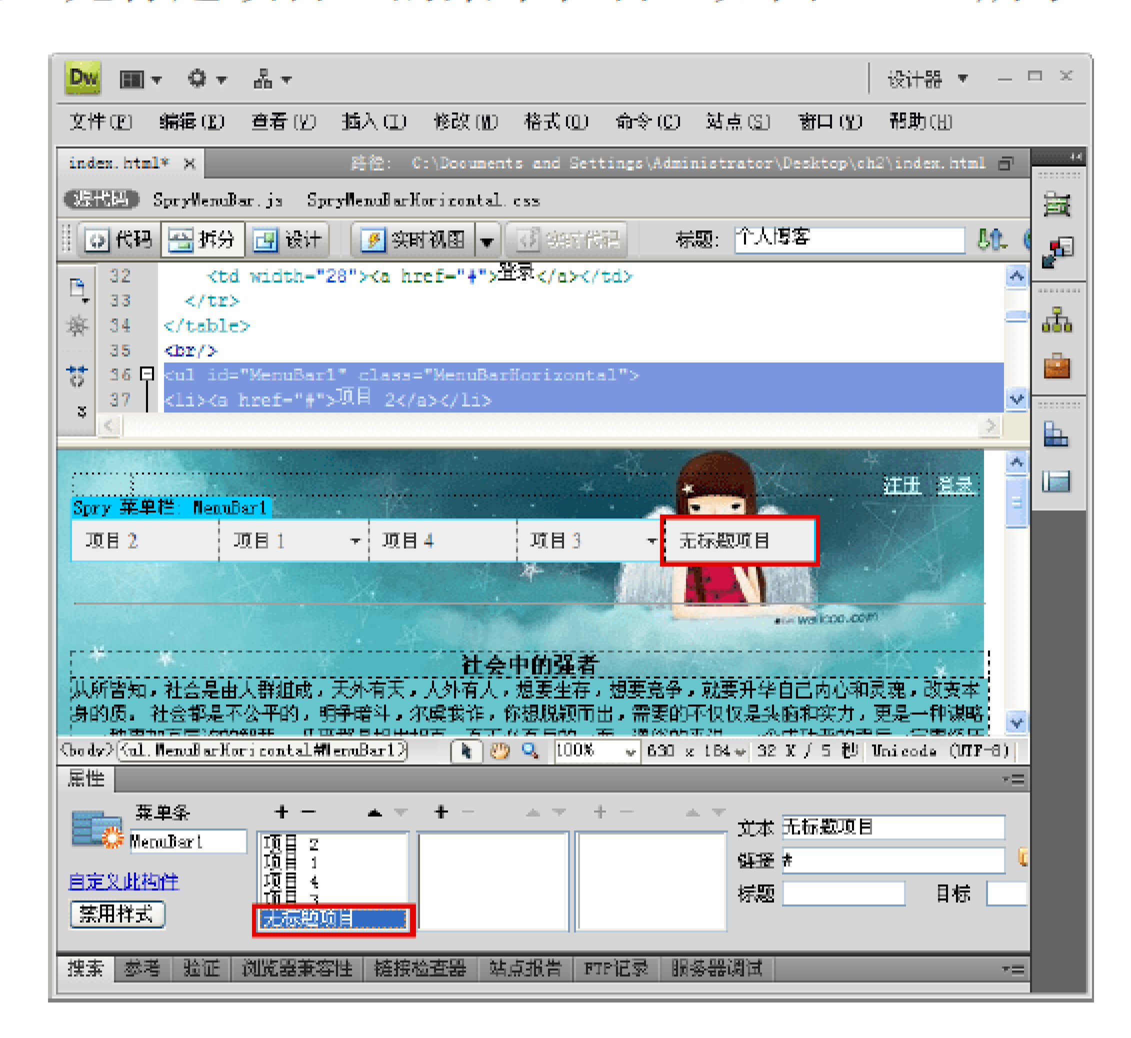


图15-36　添加主菜单项目



添加的菜单栏目没有子菜单，需要在子菜单选项中使用相同方法添加。

如果要删除某个主菜单栏目，只要选中该栏目名称，单击“删除菜单项”按钮一即可。同时，其相应的子菜单也被删除，如图15-37所示。

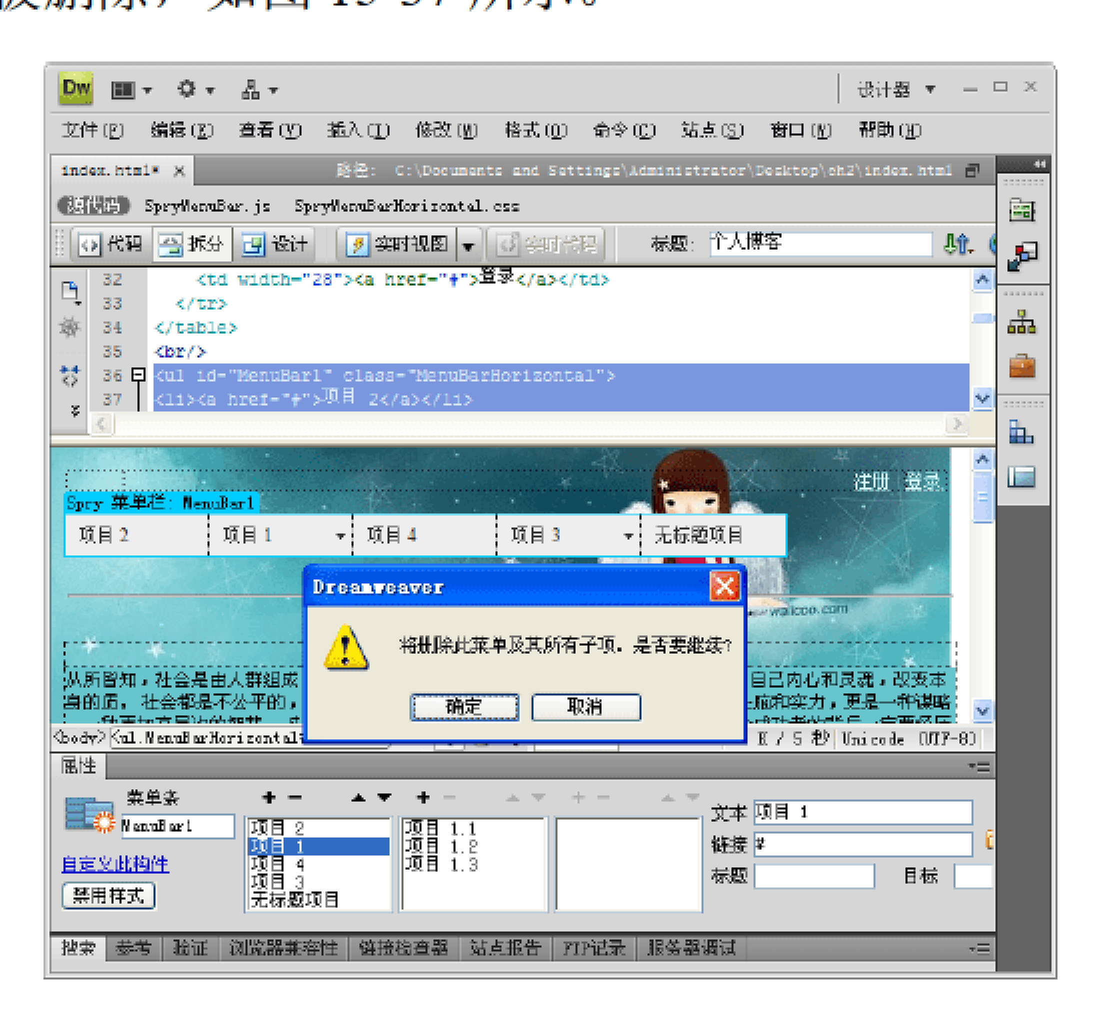


图15-37　删除主菜单栏目

在文档中只看到主菜单栏目，无法查看其相应的子菜单栏目。要想查看子菜单栏目有两种方法，一种是在“属性”面板中单击子菜单栏目名称，文档中即可显示子菜单；还有一种方法

是单击“属性”面板中的“禁用样式”按钮，文档中的菜单就会以项目列表的形式将所有菜单栏目展示出来，如图 15-38 所示。

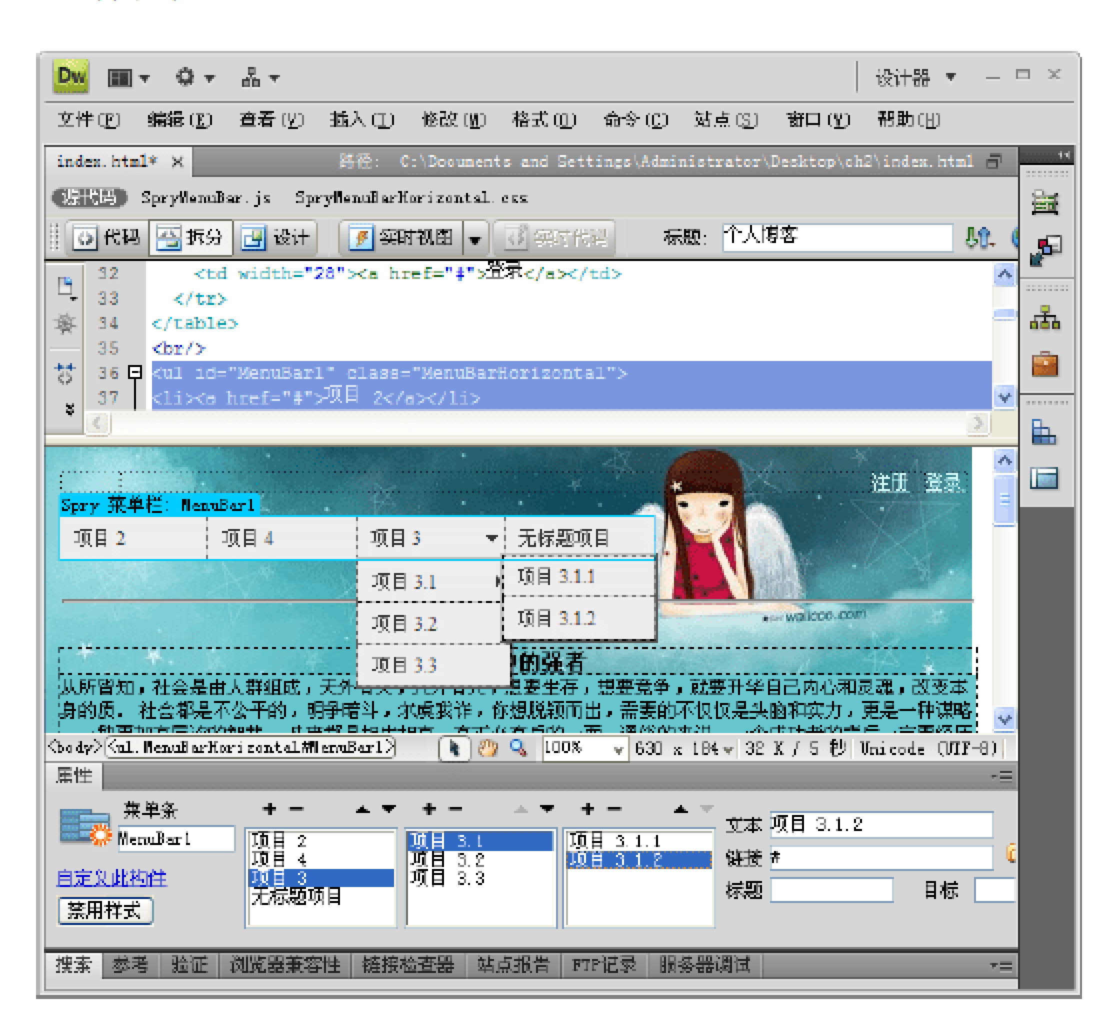


图 15-38　查看子菜单栏目

2．设置菜单栏目基本属性

插入的 Spry 菜单栏已经具备了所有功能，比如菜单栏目名称与超级链接，只需要将这些功能根据所在的网页内容具体化。方法是在“属性”面板中选中要更改的栏目名称，然后设置“文本”与“链接”选项即可，如图 15-39 所示。

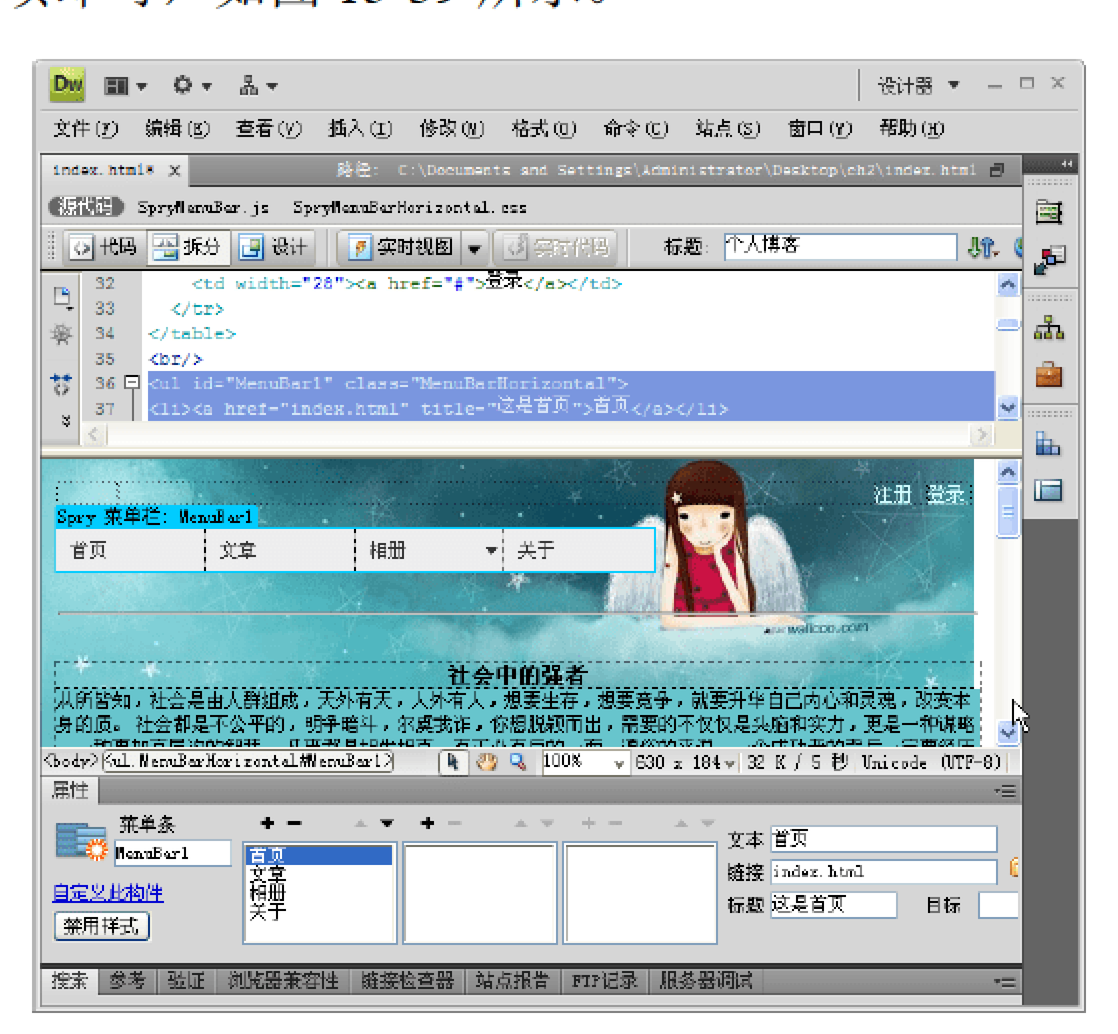


图 15-39　更改栏目名称与链接



即使具有子菜单的菜单栏目也可以设置超级链接，这样既可以通过该菜单栏目显示相应的子菜单，也可以单击该菜单栏目跳转到其他网页。

“属性”面板中的“标题”用来设置鼠标指向菜单栏目时的提示文字，如图 15-40 所示。“目标”选项用来设置链接文件的打开方式。后者在没有设置的情况下，链接文件会在同一窗

口中打开。

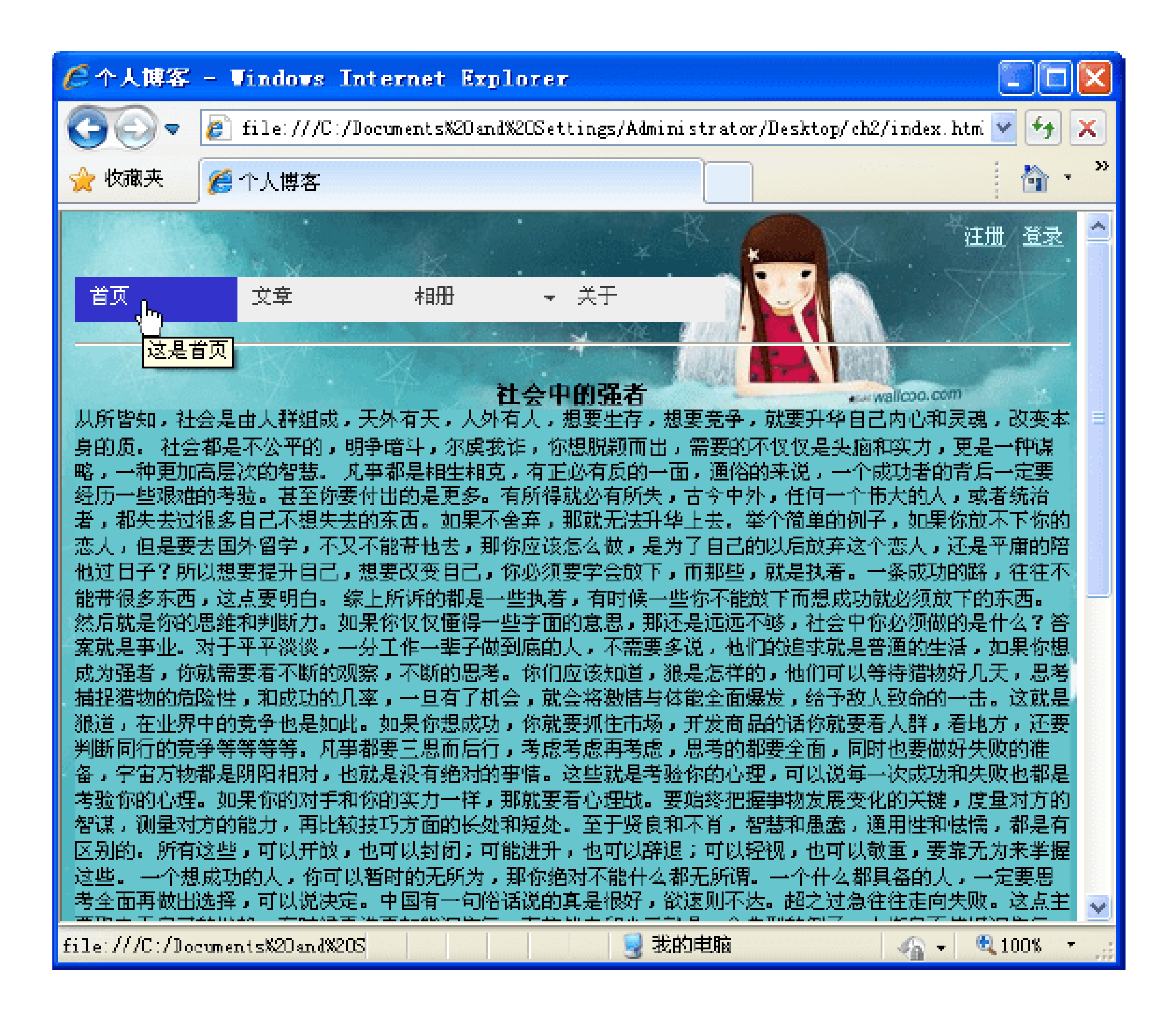


图 15-40　显示提示信息

3．定位子菜单

当创建好 Spry 菜单栏后，保存文档并且按 F12 键打开 IE 窗口预览，发现子菜单栏目与上级菜单栏目之间的上、下、左、右位置是固定的。在“CSS 样式”面板中，选中 ul.MenuBarHorizontal ul 规则，单击“编辑样式”按钮 。在“方框”选项卡中，通过设置“边界”选项中的参数来改变子菜单栏目与上级菜单栏目之间的位置。

4．更改菜单项的尺寸

在“CSS 样式”面板中双击 ul.MenuBarHorizontal li 规则，在“定位”选项卡中，设置 Width 参数改变主菜单栏目宽度；双击 ul.MenuBarHorizontal ul li 规则可以改变子菜单栏目宽度。

5．更改菜单项的背景颜色与文本样式

Spry 菜单栏中背景颜色与文本样式的状态是同步的，所以两者使用的 CSS 规则是相同的，只需设置不同的选项卡即可。

1) 默认状态

在默认情况下，菜单栏目背景颜色为浅灰色，而文本颜色为深灰色。在“CSS 样式”面板中双击 ul.MenuBarHorizontal a 规则，可以更改菜单栏目中的字体和背景颜色。



在“类型”选项卡中只是设置了“字体”参数，其他参数没有更改。当然也可以在其中设置更多的效果。

2) 鼠标指向栏目状态

双击“CSS 样式”面板中的 ul.MenuBarHorizontal a:hover, ul.MenuBarHorizontal a:focus 规则，可改变鼠标指向菜单栏目时的状态显示，例如修改为加粗并加下划线显示。

3) 上级菜单栏目状态

当鼠标指向子菜单栏目后，相应的上级菜单栏目会呈现一个显示状态，说明子菜单是在该菜

单中打开的。这种显示状态可以通过 ul.MenuBarHorizontal a.MenuBarItemHover, ul.MenuBarHorizontal a.MenuBarItemSubmenuHover, ul.MenuBarHorizontal a.MenuBarSubmenuVisible 规则设置。



Spry 组件的 CSS 文件是自动保存在站点的 SpryAssets 文件夹中的，要想设置不同样式的 Spry 组件，必须在不同的站点中创建 Spry 组件。

15.4.3　实例描述

导航对于成熟的网站来说，是一个必不可少的功能模块。本实例通过使用 Spry 框架中的菜单栏组件来完成网站导航的功能。该菜单栏有水平排列和垂直排列两种方式，本实例使用水平方式显示菜单效果。

15.4.4　实例应用

【例 15-4】首页导航

使用 Dreamweaver CS4 插入 Spry 框架中的水平菜单，方法是：单击“布局”选项卡中的“Spry 菜单栏”按钮 Spry 菜单栏，然后在“Spry 菜单栏”对话框中通过“水平”单选按钮来选择布局选项。菜单栏布局效果如图 15-41 所示。

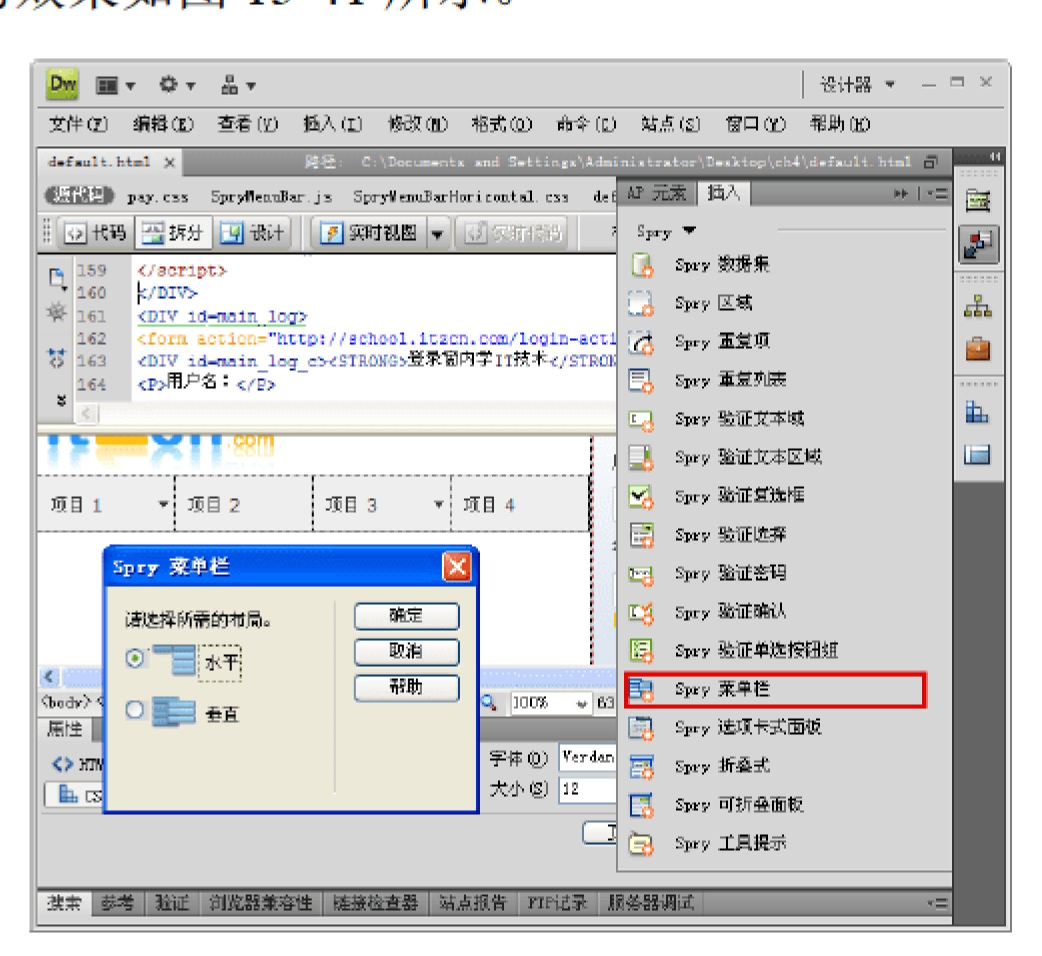


图 15-41　选择水平插入菜单

创建 Spry 菜单栏组件后，单击 Spry 菜单栏，在“属性”面板中显示菜单条属性，如图 15-42 所示。

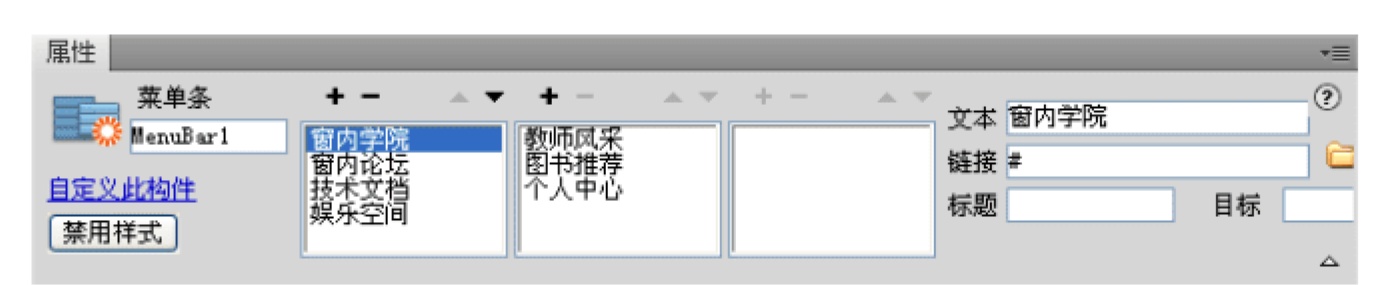


图 15-42　Spry 菜单栏的“属性”面板

可以对导航的文本进行修改、添加和删除等功能操作。通过“链接”属性可以设置选择该菜单后的转向页面。

15.4.5 运行结果

在浏览器中运行 default.html 页面，效果如图 15-43 所示。



图 15-43 首页导航

15.4.6 实例分析



源码解析：

通过以上实例，可以发现使用 Spry 框架实现菜单效果非常方便，只需要将该空间拖动到页面，然后设置相关的属性即可。在使用该控件时，每个选项都可以设置对应的链接地址。

15.5 制作导航简介

选项卡式面板组件是一组面板，用来将内容存储到紧凑的空间中。访问者可以通过单击要访问的面板上的选项卡，来隐藏或者显示在选项卡式面板中的内容。当访问者单击不同的选项卡时，组件的面板会相应地打开。



视频教学：光盘/视频/15/15.5 制作导航简介.avi 长度：5 分钟

15.5.1 基础知识——插入选项卡

要在页面中插入选项卡式面板，首先要保存当前页面。然后，从“插入”工具栏的“布局”

选项卡中单击“Spry 选项卡面板”按钮 Spry 选项卡式面板，此时在光标所在位置即可创建 Spry 选项卡面板，如图 15-44 所示。

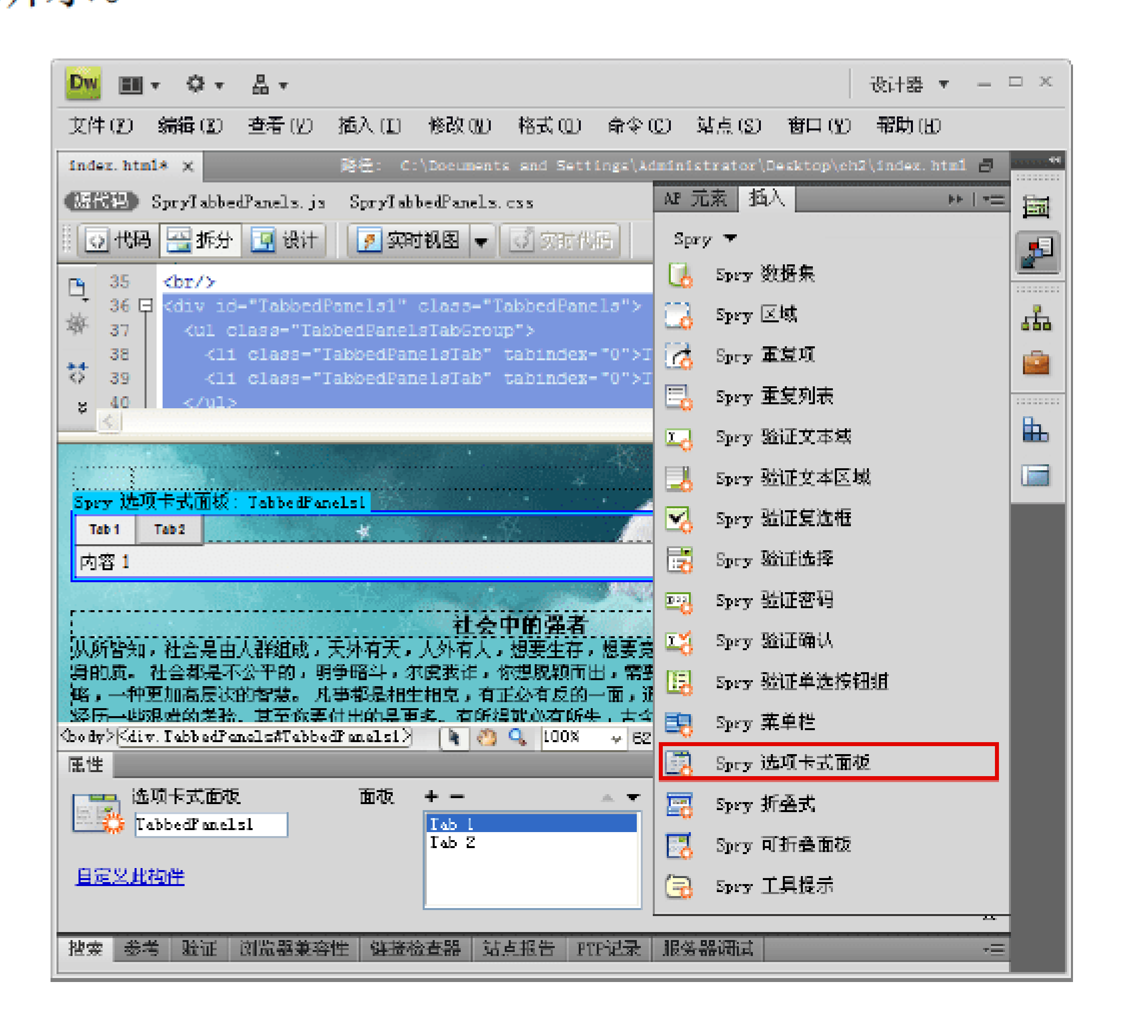


图 15-44 创建 Spry 选项卡面板

如图 15-41 所示，默认的 Spry 选项卡面板组件带有两个选项卡。选项卡式组件的 HTML 代码中包含一个含有所有面板的外部 div 标签、一个标签列表、一个用来包含内容面板的 div 以及各面板对应的 div。选项卡式面板组件的 HTML 中，在文档头和选项卡面板组件的 HTML 标记之后还包括脚本标签。

15.5.2 基础知识——编辑选项卡

插入选项卡式面板完成后，默认第一个面板 Tab1 处于打开状态，在“属性”面板中会显示相应的选项，如图 15-45 所示。

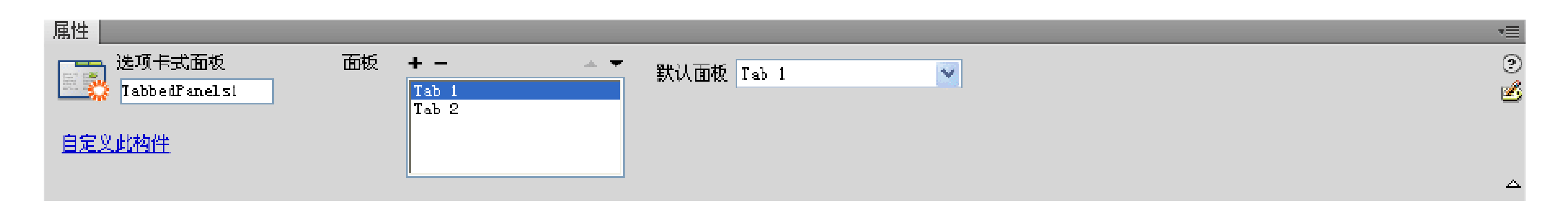


图 15-45 Spry 选项卡面板的“属性”面板

1. 设置选项卡面板基本属性

要想打开其他选项卡面板，首先将光标指向该选项卡，然后单击图标即可打开该面板。当然也可以在“属性”面板中单击选项卡的名称来打开该面板。

接着选中文本，在面板中通过输入文本，或者插入图像来更改其内容。

2. 调整选项卡面板

创建的 Spry 选项卡面板包括两个选项卡，选中选项卡面板后，在“属性”面板中选中其中一个选项，再单击“添加面板”按钮+，即可在该名称下方添加一个面板。

如果发现选项卡位置不正确，可以通过向下▼或者向上▲按钮移动选项卡的位置，使 Spry

选项卡面板布局更加合理。



默认情况下，选项卡式面板组件会展开以填充可用空间。但是，用户可以通过设置折叠式容器 width 属性来限制选项卡式面板组件的宽度。方法是在 SpryTabbedPanels.css 文件中修改.TabbedPanels 规则，该规则可为选项卡式面板组件的主容器元素定义属性，添加一个 width 属性即可，例如“width:300px”。

15.5.3 实例描述

继续完成 15.4 节中的实例，本实例在原有实例的基础上新增加一个导航简介功能。所谓的导航简介功能就是使用选项卡来实现的界面切换。该功能可以大大减少内容在页面中的占有量，同时还可以方便用户的操作。本实例中的导航简介就是使用选项卡控件来完成的。

15.5.4 实例应用

【例 15-5】导航简介

首先要保存当前页面。然后，从“插入”工具栏的“布局”选项卡中单击“Spry 选项卡式面板”按钮 Spry 选项卡式面板 ，此时在光标所在位置即可创建 Spry 选项卡面板，如图 15-46 所示。

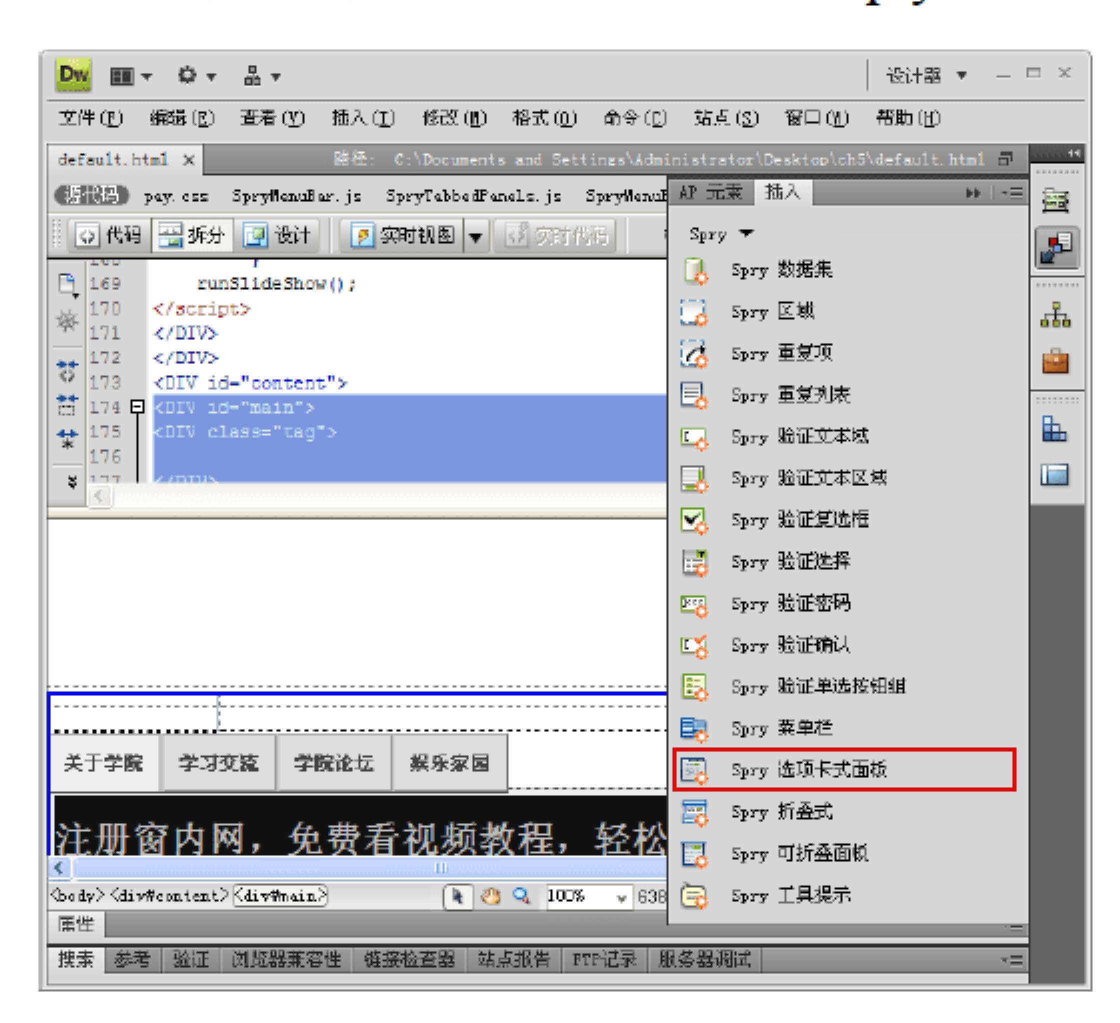


图 15-46 插入选项卡面板

接下来添加选项卡。在 Spry 面板中设置“关于学院”、“学习交流”、“学院论坛”和“娱乐家园”等选项卡，在对应的选项卡中设置需要显示的导航简介信息，如图 15-47 所示。

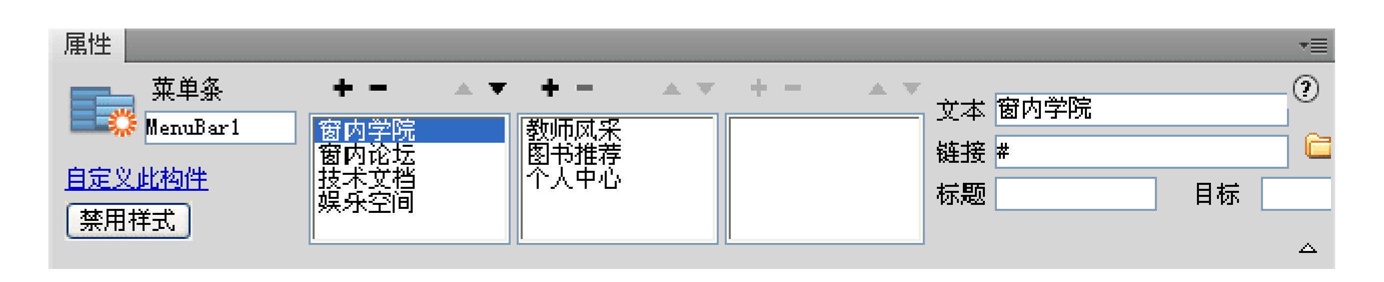


图 15-47 需要显示的导航简介信息

设置好相应的选项信息，保存页面则会在页面中生成如下代码。

```
<div id="TabbedPanels1" class="TabbedPanels">
    <ul class="TabbedPanelsTabGroup">
        <li class="TabbedPanelsTab" tabindex="0">关于学院</li>
        <li class="TabbedPanelsTab" tabindex="0">学习交流</li>
        <li class="TabbedPanelsTab" tabindex="0">学院论坛</li>
        <li class="TabbedPanelsTab" tabindex="0">娱乐家园</li>
    </ul>
    <div class="TabbedPanelsContentGroup">
        <div class="TabbedPanelsContent">关于学院导航简介内容</div>
        <div class="TabbedPanelsContent">学习交流导航简介内容</div>
        <div class="TabbedPanelsContent">学院论坛导航简介内容</div>
        <div class="TabbedPanelsContent">娱乐家园导航简介内容</div>
    </div>
</div>
```

15.5.5　运行结果

保存设置好的相关信息后，在浏览器中运行 default.html 页面，效果如图 15-48 所示。



图 15-48　首页导航简介运行结果

15.5.6　实例分析



源码解析：

在这个实例中，使用了选项卡控件完成网站导航简介功能。从该控件生成的代码中可以看到，它是以列表形式显示的，每个选项按钮都是 li 列表，而显示的面板信息则是使用 div 容器。这样方便开发人员对该空间的灵活操作。

15.6 窗 内 公 告

本实例将使用 Dreamweaver CS4 中的 Spry 折叠组件来实现。Spry 是一个可折叠面板，可以将大量内容存储在一个紧凑的空间中。访问者可以通过单击该面板上的选项卡来隐藏或者显示折叠组件中的内容。折叠面板分为两种：一种是多个面板的折叠式面板组件，一种是只有一个面板的可折叠面板组件。

**视频教学：**光盘/视频/15/15.6 窗内公告.avi 长度：5 分钟

15.6.1 基础知识——插入折叠式面板

Spry 折叠式组件是一组可折叠的面板，单击不同的选项卡时，折叠组件的面板会相应地展开或者收缩。在折叠组件中，每次只能有一个内容面板处于打开且可见的状态。

要创建 Spry 折叠式组件，只要单击“布局”选项卡中的“Spry 折叠式”按钮 Spry 折叠式 即可，如图 15-49 所示。

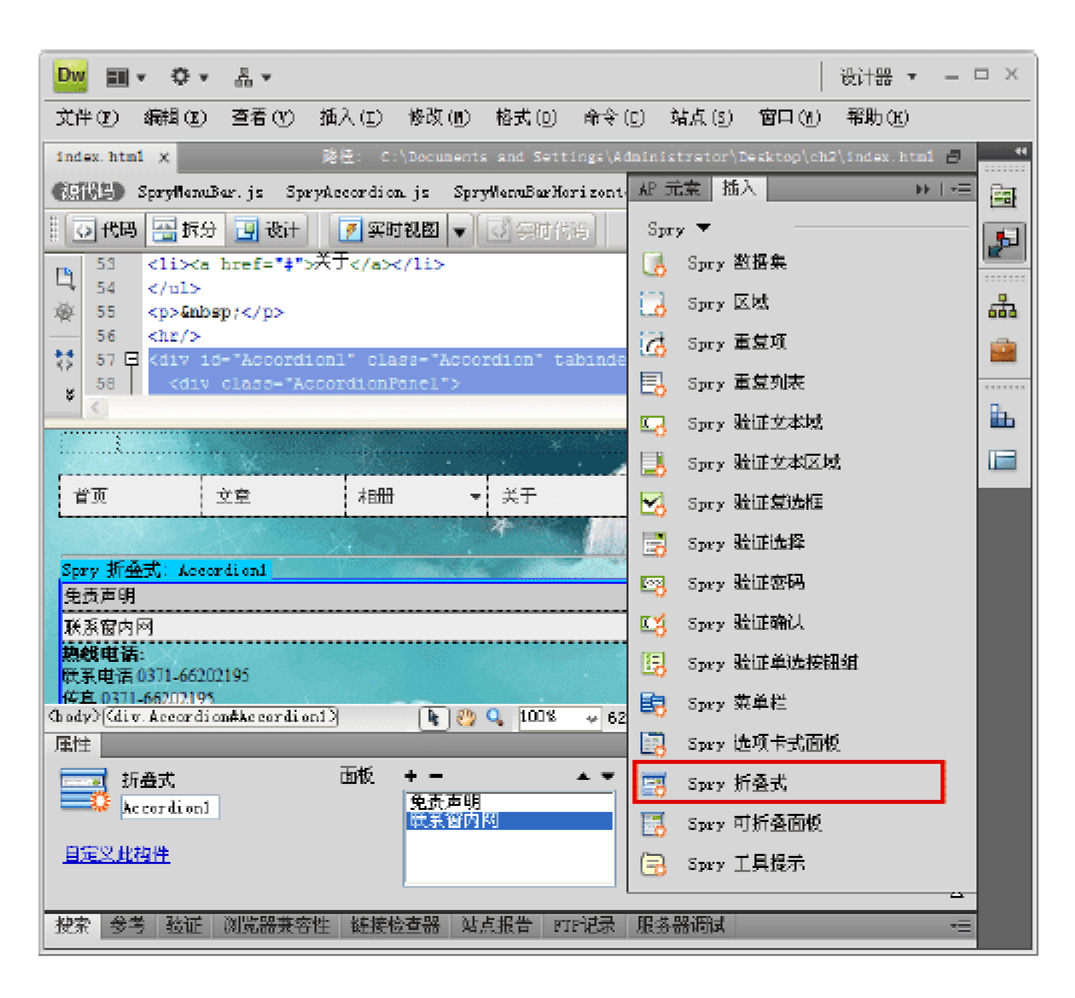


图 15-49 创建 Spry 折叠式组件

Spry 折叠式组件的编辑方法与 Spry 选项卡面板基本相同，只是前者只能添加或者删除面板，以及调整面板顺序，不能更改预览中默认打开的面板，如图 15-50 所示。在预览中发现，面板高度不会随着面板中的内容变化。

折叠式组件的默认 HTML 中包括所有面板的外部 div 标签以及各面板对应的 div 标签。各面板的标签中还有一个标题 div 和内容 div。折叠式组件可以包含任意数量的独立面板。在折叠式组件的 HTML 中还包括 script 标签。

下面介绍 Spry 可折叠面板组件的插入方法。Spry 可折叠面板组件可以将内容存储到紧凑的空间中，单击组件的选项卡即可隐藏或者显示可折叠面板中的内容。



图 15-50　预览 Spry 折叠式面板

方法是：单击“布局”选项卡中的“Spry 可折叠面板”按钮 Spry 可折叠面板，创建 Spry 可折叠面板，其中“属性”面板显示如图 15-51 所示。

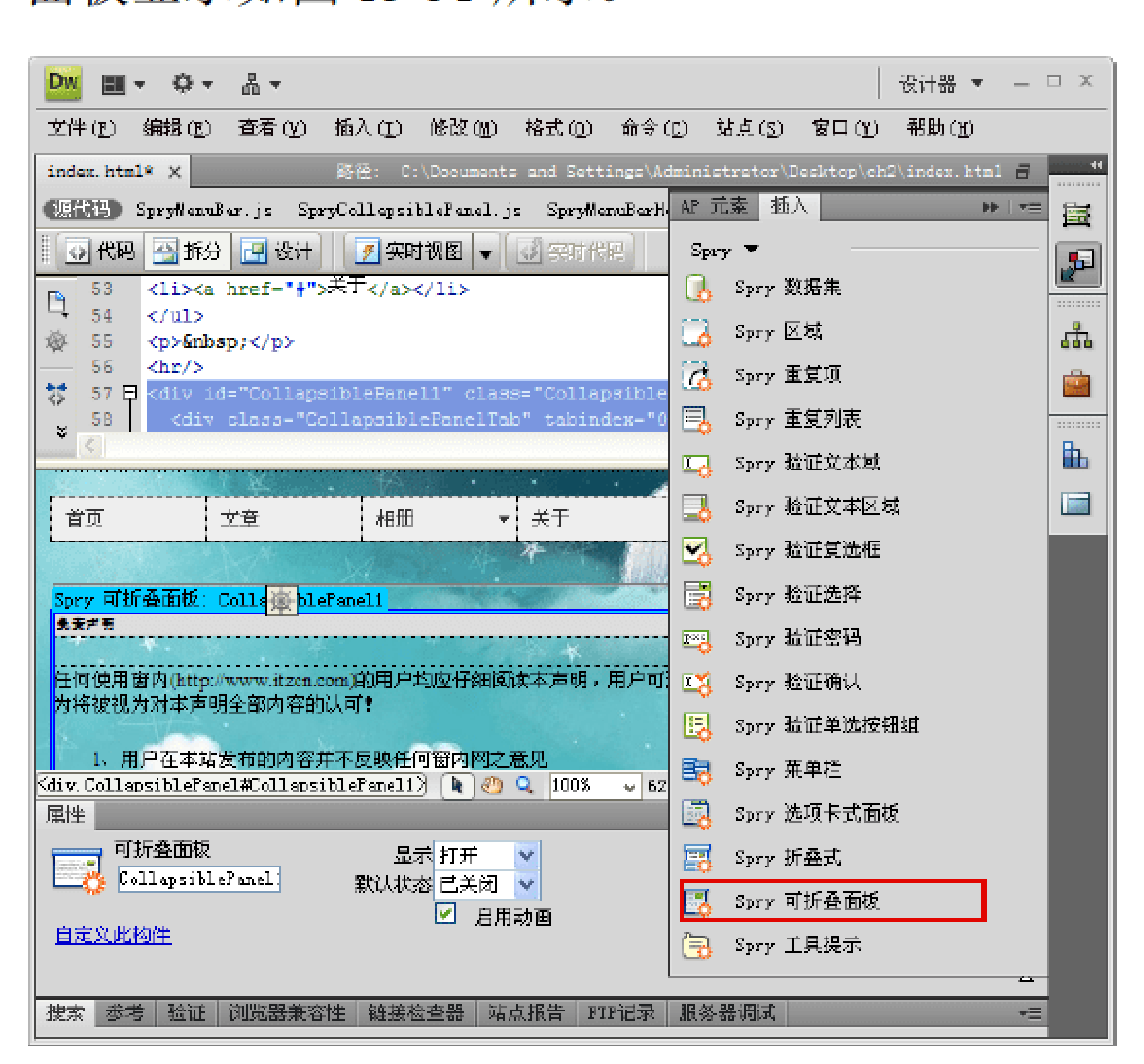


图 15-51　创建 Spry 可折叠面板

Spry 可折叠面板虽然只有一个面板，在文档中还是可以显示或者隐藏面板内容。方法是将鼠标指向该面板的选项卡，然后单击“眼睛”图标即可。或者直接在“属性”面板中，选择“显示”下拉列表中的“打开”或者“已关闭”选项，如图 15-52 所示。

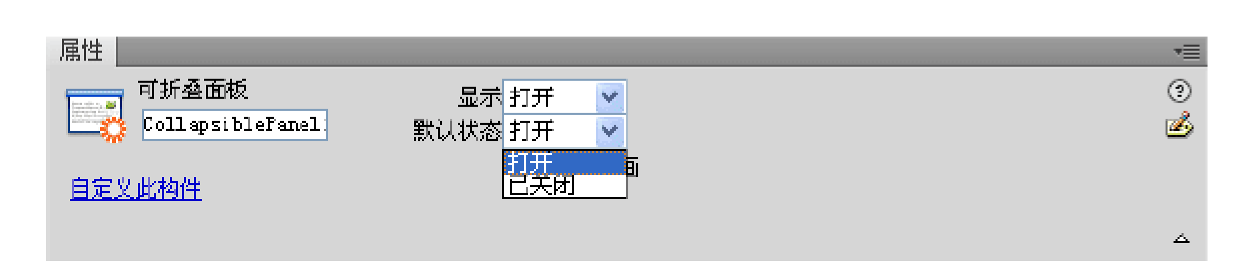


图 15-52　在文档中显示或者隐藏面板内容

在“属性”面板中，还可以设置预览网页时的默认状态。选择“默认状态”下拉列表中的“打开”或者“已关闭”选项，即可在打开的网页中查看打开或者已经关闭的面板效果。图 15-53 所示为默认状态为已关闭的 Spry 可折叠面板。

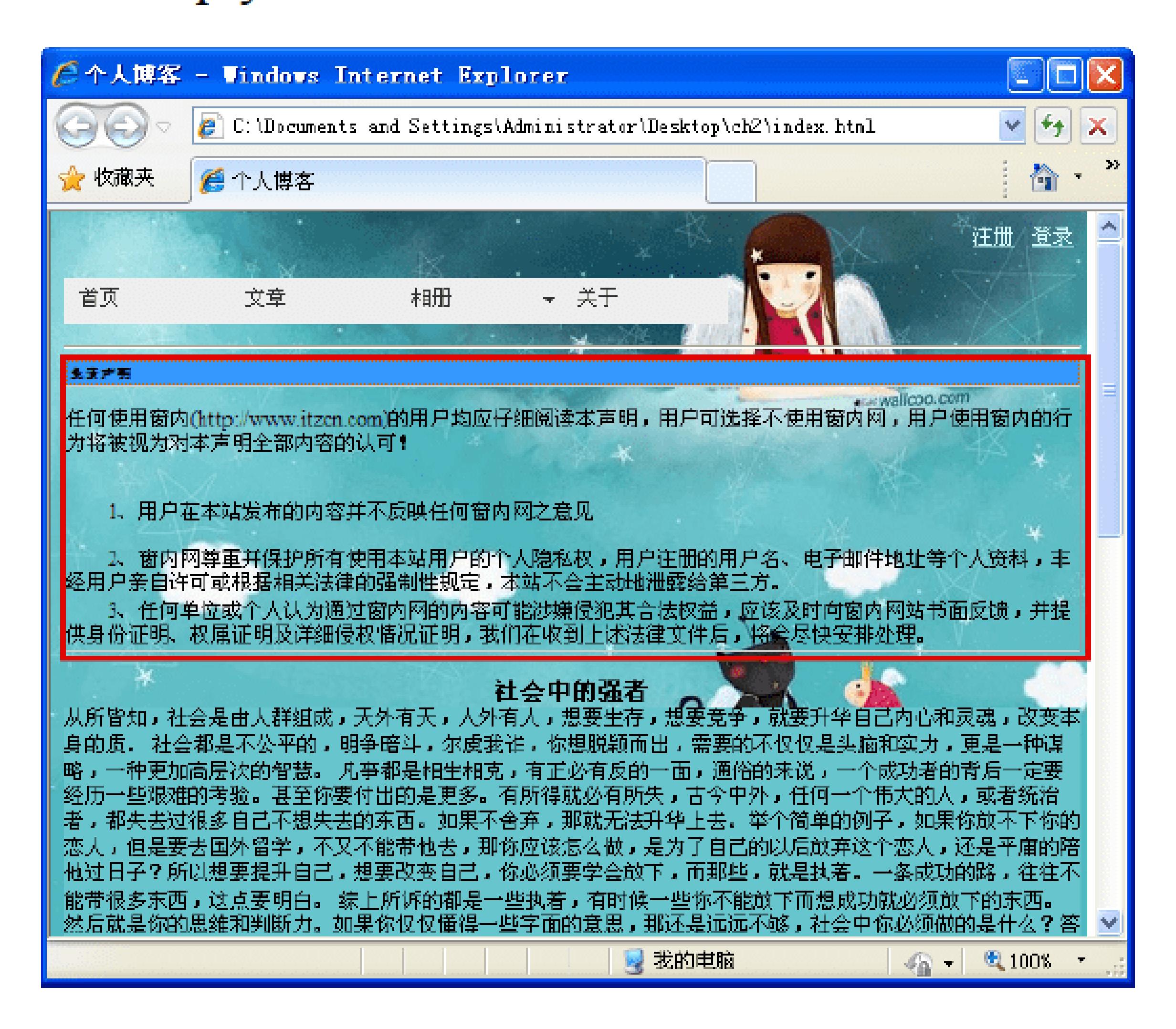


图 15-53　预览 Spry 可折叠面板



Spry 可折叠面板在打开或者关闭面板时具有动画效果，这是因为在“属性”面板中启用了“启用动画”选项。如果禁用了该选项，则可折叠面板会迅速打开和关闭。

15.6.2　基础知识——自定义折叠面板

在 3 种 Spry 组件中，Spry 选项卡式面板、Spry 折叠式与 Spry 可折叠面板虽然展示效果不同，但是组件样式基本相同，例如面板的宽度、背景颜色以及内部的文本样式等。下面以 Spry 折叠式组件为例介绍如何更改组件样式。

1．限制面板宽度

默认情况下，折叠组件会展开以填充可用空间。当 IE 窗口的宽度发生变化时，折叠组件的宽度会随之变化。要想改变折叠组件的宽度有两种方法，一种是将折叠组件建立在固定表格中，通过表格的宽度限制折叠组件的宽度；另一种是真正地限制其宽度，即通过设置折叠式容器的 width 属性来限制折叠组件的宽度。

修改方法是：选中 Spry 折叠组件后，在“CSS 样式”面板中选中元素的“.Accordion”CSS 规则。然后单击“编辑样式”按钮，打开“.Accordion 的 CSS 规则定义”对话框，在“定义”选项中设置“宽”参数即可。



在“CSS 样式”面板中的“全部”选项卡中双击 Accordion 规则，也可以打开相同的对话框设置折叠组件宽度。

设置新的宽度值后，保存 SpryAccordion.css 文件并且返回折叠组件所在文档。再到 IE 窗

口中预览，发现无论 IE 窗口的宽度如何变化，折叠面板的宽度都不会发生变化。



当设置 CSS 规则并且关闭对话框后，会弹出 SpryAccordion.css 文件。必须保存该文件后才能在 IE 窗口中查看更改后的效果。

2. 更改折叠组件的背景颜色

在预览 Spry 折叠式面板时，发现该面板具有不同状态中的不同背景颜色。要更改不同部分的背景颜色，可以使用表 15-2 所提供的参数进行设置。

表 15-2　Spry 折叠式组件中的背景颜色 CSS 规则

要更改的组件部分	相关 CSS 规则	要添加或更改的属性和值的实例
折叠式面板选项卡的背景颜色	.AccordionPanelTab	background-color: #CCCCCC;(默认值)
折叠式内容面板的背景颜色	.AccordionPanelContent	background-color: #CCCCCC; (默认值)
已打开的折叠式面板的背景颜色	.AccordionPanelOpen .AccordionPanelTab	background-color: #EEEEEE; (默认值)

在折叠式组件中，还包括鼠标响应背景颜色事件。单击某个选项卡后，该选项卡面板处于打开状态并且其背景颜色为湖蓝色(#33CCFF)；而其他选项卡背景颜色为蓝色(#3399FF)。这两个背景颜色分别在.AccordionFocused .AccordionPanelOpen .AccordionPanelTab 与.AccordionFocused .AccordionPanelTab 规则中设置。

3. 设置折叠组件文本的样式

Spry 折叠式组件中的文本分为选项卡文本与面板中的文本，所以在设置文本样式时，除了这两种文本样式可以分别设置外，还可以总体设置整个组件中的文本。表 15-3 列出了不同部分的文本所在的 CSS 规则。

表 15-3　不同部分的文本相关 CSS 规则

要更改的文本	相关 CSS 规则	要添加的属性和值的实例
整个折叠组件(包括选项卡和内容面板)中的文本	.Accordion 或 .AccordionPanel	font: Arial; font-size:medium;
仅限折叠式面板选项卡中的文本	.AccordionPanelTab	font: Arial; font-size:medium;
仅限折叠式内容面板中的文本	.AccordionPanelContent	font: Arial; font-size:medium;



Spry 选项卡式面板、Spry 可折叠面板、Spry 折叠式的设置基本相同，只是由于不同的组件，其名称各不相同。Spry 选项卡式面板名称为 TabbedPanels；Spry 可折叠面板名称为 CollapsiblePanel。

15.6.3 实例描述

我们将以“窗内网公告”模块为例，向大家介绍使用 Spry 中的折叠组件来开发 Web 功能的方法。该实例主要的功能是向用户展示站内公告信息或者站内的当前状态信息，这样可以方便用户直观地了解该站的当前内容。

15.6.4 实例应用

【例 15-6】窗内公告

创建 Spry 折叠式组件。单击“布局”选项卡中的“Spry 折叠式”按钮 Spry 折叠式 添加折叠式组件，详细如图 15-54 所示。

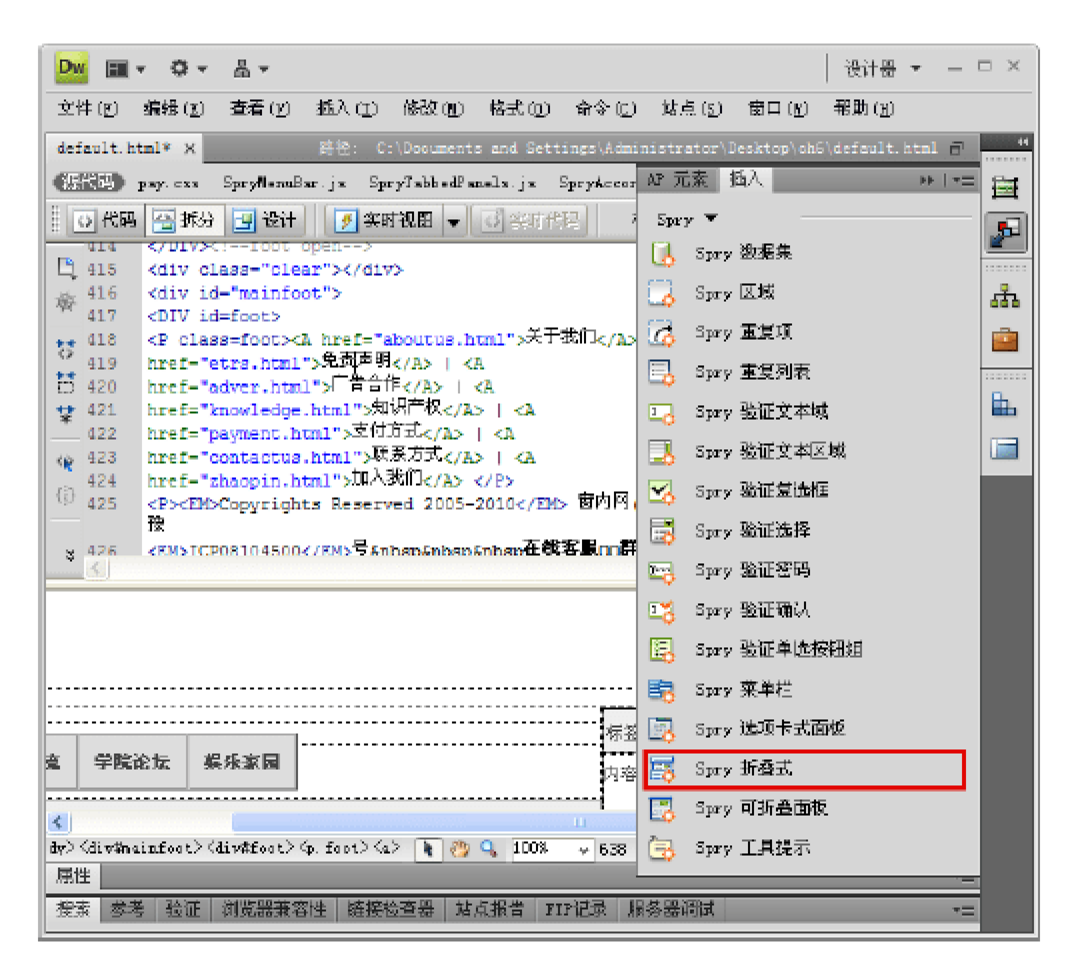


图 15-54 添加折叠组件

该组件的编辑方法与 Spry 选项卡面板基本相同，接下来设置该组件的属性信息，如图 15-55 所示。



图 15-55 折叠组件属性设置

配置好相关的折叠组件的属性后则需要对折叠面板添加内容，最后生成代码如下：

```
<div id="Accordion1" class="Accordion" tabindex="0">
   <div class="AccordionPanel">
     <div class="AccordionPanelTab">窗内公告</div>
     <div class="AccordionPanelContent">
     网络安全视频教程 Oracle 1<br/>
     1g 完全学习手册视频教程 Windows XP<br/>
```

```
        视频教程 Dreamweaver <br/>
        视频教程 Flash 视频教程 Word 2007</div>
    </div>
    <div class="AccordionPanel">
      <div class="AccordionPanelTab">窗内动网</div>
      <div class="AccordionPanelContent"><UL>
  <script src="http://bbs.itzcn.com/api/javascript.php?key=threads_school_
index right" type="text/javascript"></script></UL></div>
    </div>
  </div>
```

15.6.5　运行结果

保存相关属性和代码后，在浏览器中运行 default.html 页面，效果如图 15-56 所示。



图 15-56　首页右侧模块运行效果

15.6.6　实例分析



源码解析：

在这个实例中使用了 Spry 组件中的折叠式组件，大大地方便了开发人员的操作，以前在做一个折叠特效时，需要花费一定的时间，而现在使用折叠效果组件只需将组件拖曳到页面上，写入需要显示的内容即可。

15.7　用 户 注 册

Dreamweaver CS4 中内置的 Spry 框架提供了丰富用户交互功能的组件。这些组件可以让用户显示/隐藏页面元素、改变呈现方式以及菜单交互等。其中的 Spry 表单组件主要用于验证用户在对象域中所输入的内容，是否为有效的数据。并且在这些对象域中，内建了 CSS 样式和

JavaScript 特效，更加丰富了对象域的显示效果。

 **视频教学：**光盘/视频/15/15.7　用户注册.avi　　　　长度：7 分钟

15.7.1　实例描述

本实例使用 Dreamweaver CS4 内置的 Spry 框架提供的组件来完成用户注册功能。该注册功能包含用户名、用户密码、密码确认、性别和出生地等功能。我们将使用 Spry 框架来实现用户名空值验证、密码框密码强度和确认密码验证等功能。

15.7.2　实例应用

【例 15-7】用户注册

Spry 验证文本域组件是一个文本域，该域用于在用户输入文本时显示文本的状态(有效或无效)。例如，可以在用户输入电子邮件地址的表单中添加验证文本域组件。如果访问者无法在电子邮件地址中输入“@”符号和句点，验证文本域组件会返回一条消息，声明用户输入的信息无效。

1．插入 Spry 验证文本域

首先在页面上单击鼠标左键，再从“插入”工具栏中的“表单”面板上单击“Spry 验证文本域”按钮 Spry 验证文本域 ，打开“输入标签辅助功能属性”对话框。在“标签”文本框中输入“您的用户名”文字，然后单击“确定”按钮在页面上插入该对象域，如图 15-57 所示。

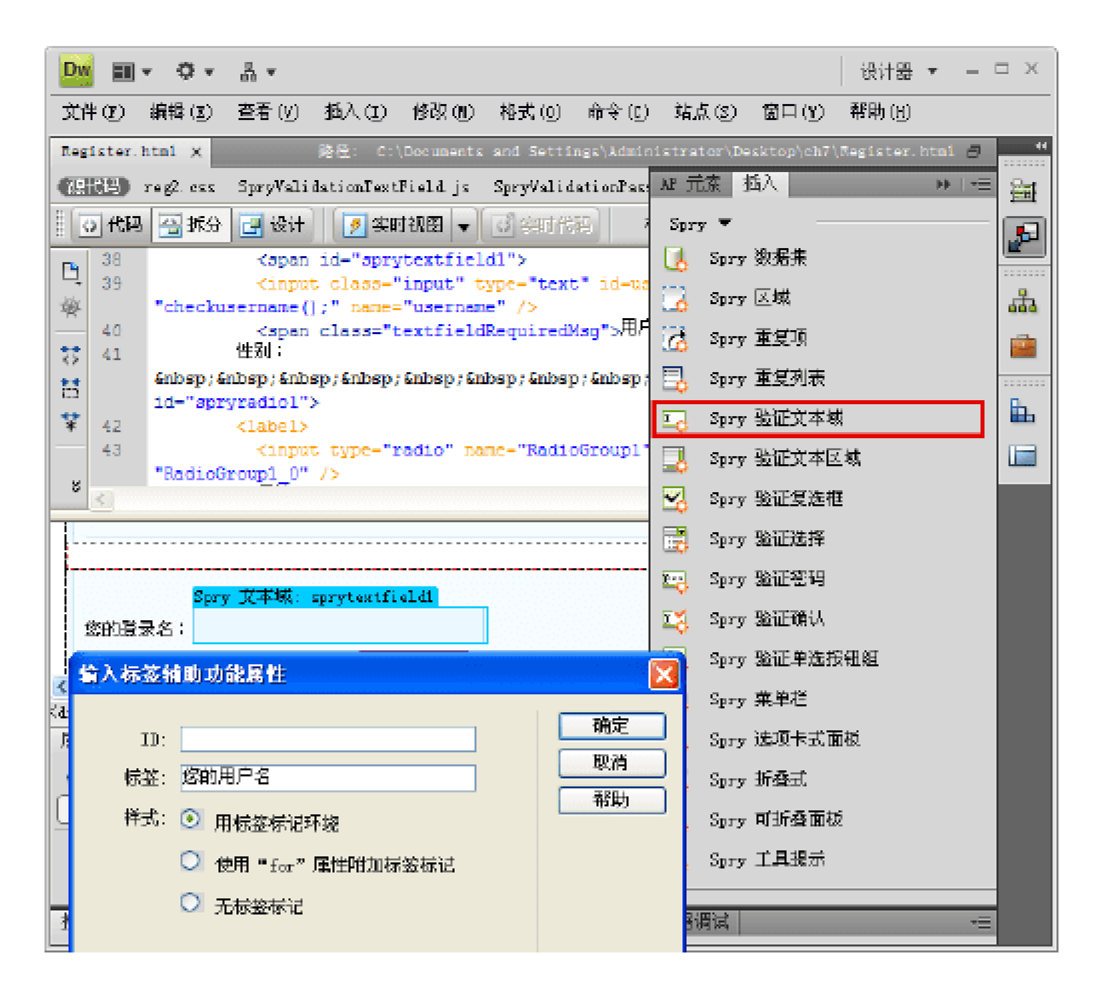


图 15-57　插入验证文本域

验证文本域组件具有许多状态(例如，有效、无效和必须值等)。用户可以根据所需的验证结果，使用“属性”面板来修改这些状态的属性，如图 15-58 所示。

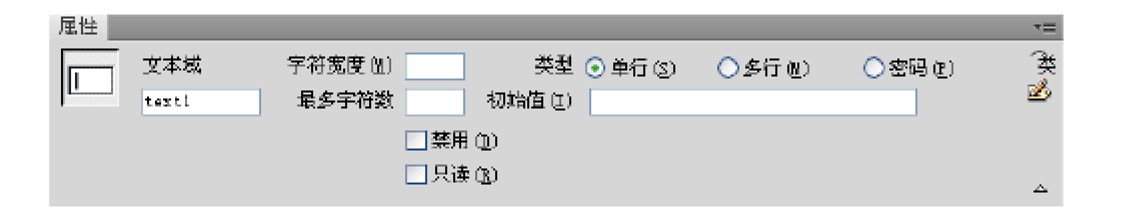


图 15-58　文本域的“属性”面板

在“属性”面板中有很多选项可以设置验证文本域的属性，在表 15-4 中列出了它们的名称及功能。

表 15-4　Spry 文本域的属性参数

名　称		功　能
Spry 文本域		标识文本域的名称。该名称也是文本域的 ID 号
类 型		确定该文本域需要验证的类型
预览状态	初始状态	在浏览器中加载页面或者用户重置表单时组件的状态
	必需状态	当用户在文本域中没有输入必需文本时组件的状态
	有效状态	当用户正确地输入信息且表单可以提交时组件的状态
	焦点状态	当用户在组件中放置插入点时组件的状态
	无效状态	当用户所输入文本的格式无效时组件的状态。例如，用 09 而不是用 2009 表示年份
验证于	onBlur	可以设置验证发生的时间(包括用户在组件外部单击时、输入内容时或尝试提交表单时)。该项为“模糊”状态，即当用户在文本域的外部单击时验证
	onChange	该项为“更改”状态，即当用户更改文本域中的文本时验证
	onSubmit	该项为“提交”状态，即当用户尝试提交表单时验证
最小字符数状态		当用户输入的字符数少于文本域所要求的最小字符数时组件的状态
最大字符数状态		当用户输入的字符数多于文本域所允许的最大字符数时组件的状态
最小值状态		当用户输入的值小于文本域所需的值时组件的状态。适用于整数、实数和数据类型验证
最大值状态		当用户输入的值大于文本域所允许的最大值时组件的状态。适用于整数、实数和数据类型验证
强制模式		用户可以禁止在验证文本域组件中输入无效字符

每当验证文本域组件以用户交互方式进入其中一种状态时，Spry 框架逻辑会向该组件的 HTML 容器添加特定的 CSS 类。例如，如果用户尝试提交表单，但尚未在必填文本域中输入文本，Spry 会向该组件应用一个类，使它显示“需要提供一个值”错误消息。

在设置过程中，用户可以为验证文本域组件指定不同的验证类型。例如，如果文本域将接收信用卡号，则可以指定信用卡验证类型。并且大多数验证类型都会使文本域要求采用标准格式，通过表 15-5 来详细了解一下文本域的格式种类。

表 15-5　文本域的验证类型

验证类型	格　式
无	无需特殊格式
整数	文本域仅接受数字

续表

验证类型	格　式
电子邮件	文本域接受包含@和句点(.)的电子邮件地址，而且@和句点的前面和后面都至少有一个字母
日期	格式可变。可以从“属性”面板的“格式”下拉列表中选择
时间	格式可变。可以从“属性”面板的“格式”下拉列表中选择。tt 表示 am/pm 格式，t 表示 a/p 格式
信用卡	格式可变。可以选择接受所有信用卡，或者指定特定种类的信用卡(MasterCard、Visa 等)。文本域不接受包含空格的信用卡号
邮政编码	格式可变。从“属性”面板的“格式”下拉列表中选择
电话号码	文本域接受美国和加拿大格式，即 (000)000-0000 或自定义格式的电话号码。如果选择自定义格式，请在“模式”文本框中输入格式，例如，000.00(00)
社会安全号码	文本域接受 000-00-0000 格式的社会安全号码
货币	文本域接受 1,000,000.00 或者 1.000.000,00 格式的货币
实数/科学记数法	验证各种数字：数字(例如 1)、浮点值(例如，12.123)、以科学记数法表示的浮点值。例如，1.212e+12、1.221e-12，其中 e 用作 10 的幂
IP 地址	格式可变。从“属性”面板的“格式”下拉列表中选择
URL	文本域接受 http://xxx.xxx.xxx 或 ftp://xxx.xxx.xxx 格式的 URL
自定义	可用于指定自定义验证类型和格式。在属性检查器中输入格式模式，并根据需要输入提示

在“属性”面板中，设置“类型”为“电子邮件地址”；输入“提示”为“输入邮箱地址”；选中 onBlur 复选框，如图 15-59 所示。

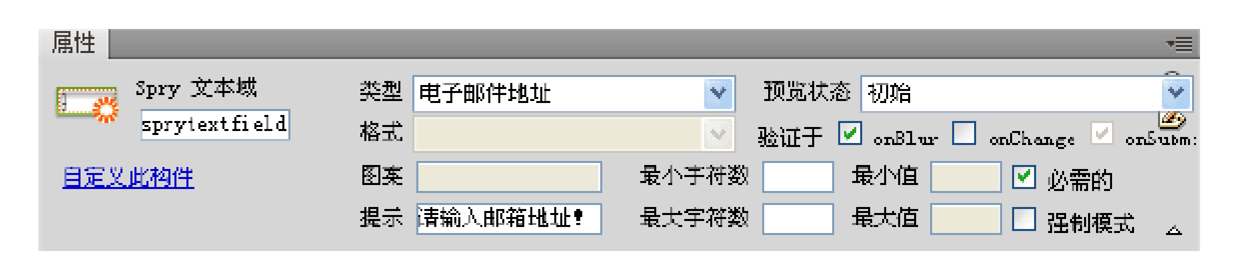


图 15-59　设置属性



用户需要选择 Spry 文本域时，可以单击表单区域上面的“Spry 文本域：sprytextfield1”选项卡，即可弹出“属性”面板。

然后，按 F12 键预览其效果。例如，在预览页面中输入错误的 E-mail 地址，即可提示“格式无效”错误信息，且以“红色”高亮度显示，如图 15-60 所示。输入正确的 E-mail 地址时，将以“绿色”显示，如图 15-61 所示。



用户可以禁止在验证文本域组件中输入无效字符。例如，如果对具有“整数”验证类型的组件集启用“强制模式”复选框。那么，当输入字母时，文本域中将不显示任何内容。

验证文本域组件的默认 HTML 通常位于表单内部，其中包含一个<span>标签，该标签将文本域的<input>标签括起来。在验证文本域组件的 HTML 中，在文档头中和验证文本域的

HTML 标记之后还包括脚本标签。



图 15-60　格式错误提示



图 15-61　邮箱正确提示

2. 修改提示信息

除此之外，用户还可以在“代码”视图中，修改提示信息。例如，将“格式无效”修改为“输入格式错误”等，代码如下所示：

```
<form id="form1" name="form1" method="post" action="">
  <span id="sprytextfield1">
  <label>Email
    <input type="text" name="text1" id="text1" />
  </label>
  <span class="textfieldRequiredMsg">需要提供一个值。
</span><span class="textfieldInvalidFormatMsg">格式无效。</span></span>
</form>
<script type="text/javascript">
<!--
var sprytextfield1 = new Spry.Widget.ValidationTextField("sprytextfield1",
"email", {hint:"请输入邮箱地址！", validateOn:["blur"]});
//-->
</script>
```

在上述代码中，只需要将代码中的“格式无效”修改为“输入格式错误”，就可以使用了。

3. 修改样式

尽管使用“属性”面板可以简化对验证文本域组件的编辑，但是并不支持自定义的样式设置任务。用户可以修改验证文本域组件的 CSS，以便根据自己的喜好创建带有样式的组件。

通过上述设置，我们了解到验证文本域组件的错误消息会以红色显示，文本周围有 1 个像素宽的边框。如果用户需要更改 CSS 规则，可以先来了解如表 15-6 所示的文本样式属性。

表 15-6　文本样式 CSS 规则

要更改的文本	相关 CSS 规则
错误消息文本	.textfieldRequiredState .textfieldRequiredMsg、.textfieldInvalidFormatState .textfieldInvalidFormatMsg、.textfieldMinValueState .textfieldMinValueMsg、.textfieldMaxValueState .textfieldMaxValueMsg、.textfieldMinCharsState .textfieldMinCharsMsg 或 .textfieldMaxCharsState .textfieldMaxCharsMsg

具体更改规则的方法如下：

先打开“Spry 文本域: sprytextfield1”选项卡，在“CSS 样式”面板中切换到“全部”选项卡。然后选择“所有规则”列表中的“.textfieldRequiredMsg, .textfieldInvalidFormatMsg, .textfieldMinValueMsg, .textfieldMaxValueMsg, .textfieldMinCharsMsg, .textfieldMaxCharsMsg, .textfieldValidMsg”规则，再单击“编辑样式”按钮。

在弹出对话框中选择“分类”列表的“类型”选项，并设置 Font-family 为“宋体 times New Roman”；Font-style 为“italic”，再选中 Text-decoration 中的“line-through”复选框。最后单击“确定”按钮返回。此时，按 F12 键预览所设置的错误消息，并显示用户所设置的文本格式，如图 15-62 所示。

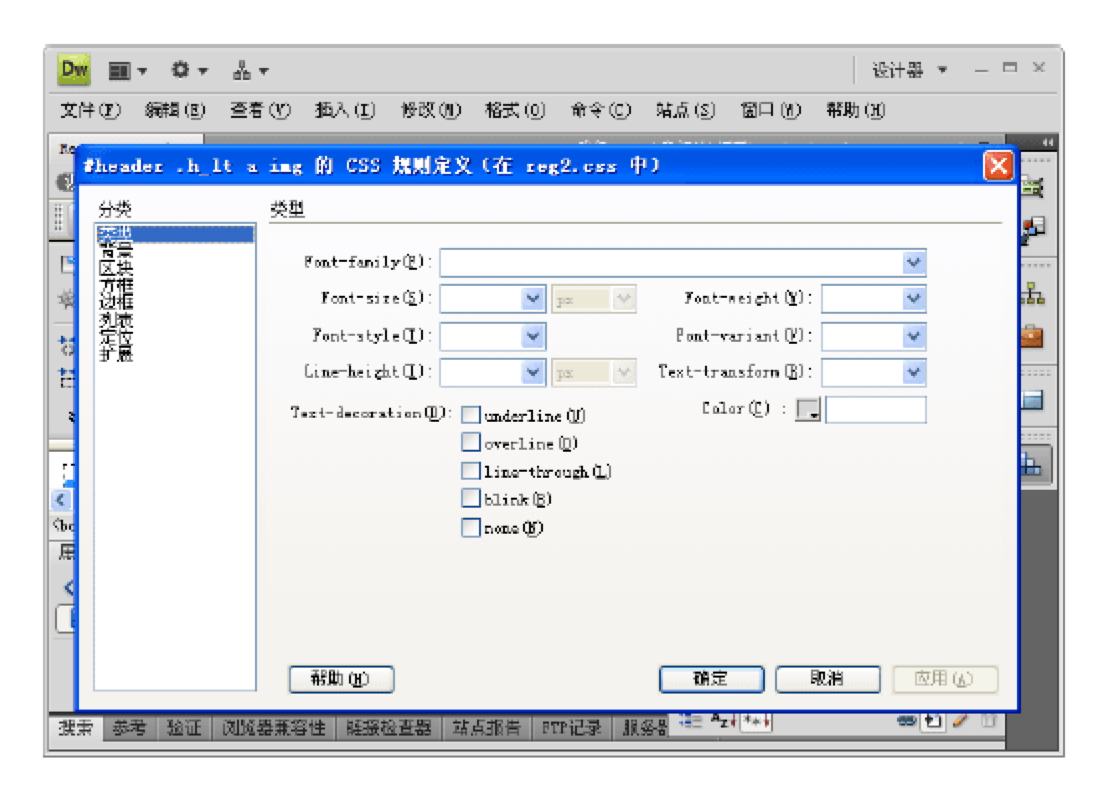


图 15-62　修改错误消息样式



如果用户需要设置文本的颜色，可以在.textfieldMaxCharsState.textfieldMaxCharsMsg 规则中，修改消息文本的字体颜色等。

Spry 验证密码组件是一个密码文本域，可用于强制执行密码规则(例如，字符的数量和类型)，组件会根据用户的输入提示警告和错误消息。在使用 Spry 验证密码组件之前，读者首先应该熟悉 Spry 验证组件的使用，如 Spry 验证文本域或者 Spry 验证文本区域。

Spry 验证密码组件可以通过“Spry 验证密码”按钮 Spry 验证密码 激活。单击它会弹出“输入标签辅助功能属性”对话框。在“标签”文本框中输入“密码”，单击“确定”按钮即可在页面中插入 Spry 验证密码组件，如图 15-63 所示。

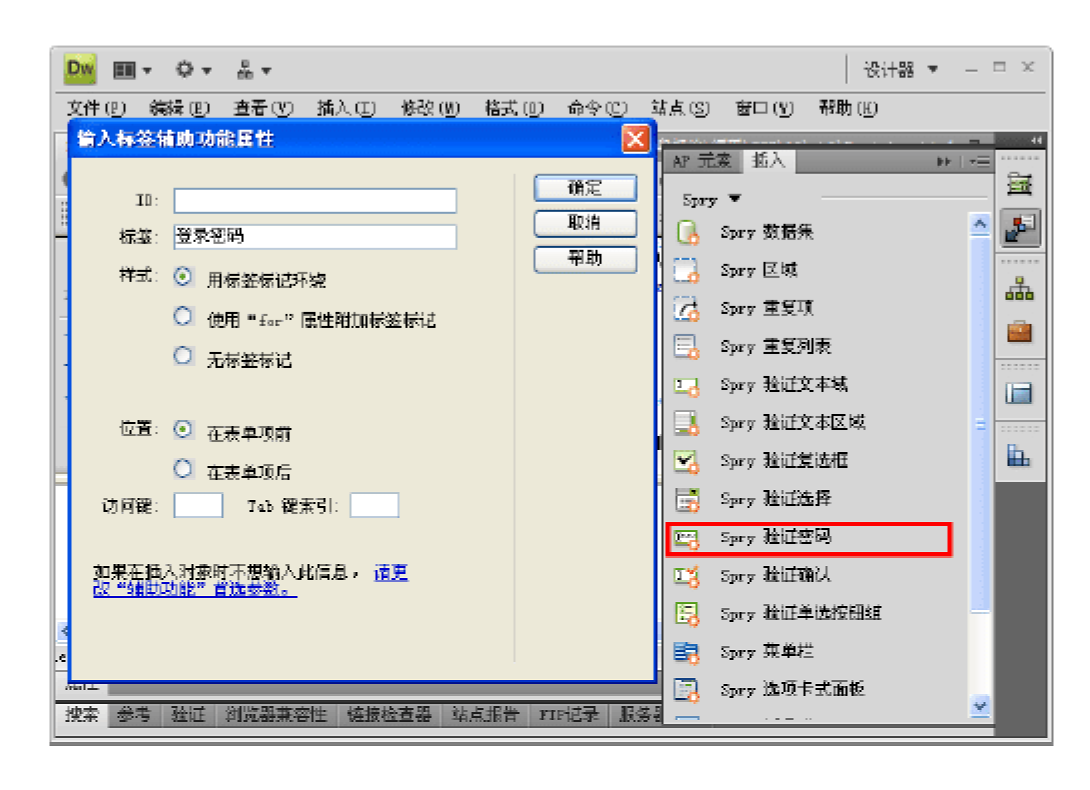


图 15-63　插入 Spry 验证密码

Spry 验证密码组件具有许多状态(例如，有效、必需和最小字符数等)。用户可以根据所需的验证结果，使用“属性”面板来修改这些状态的属性，如图 15-64 所示。

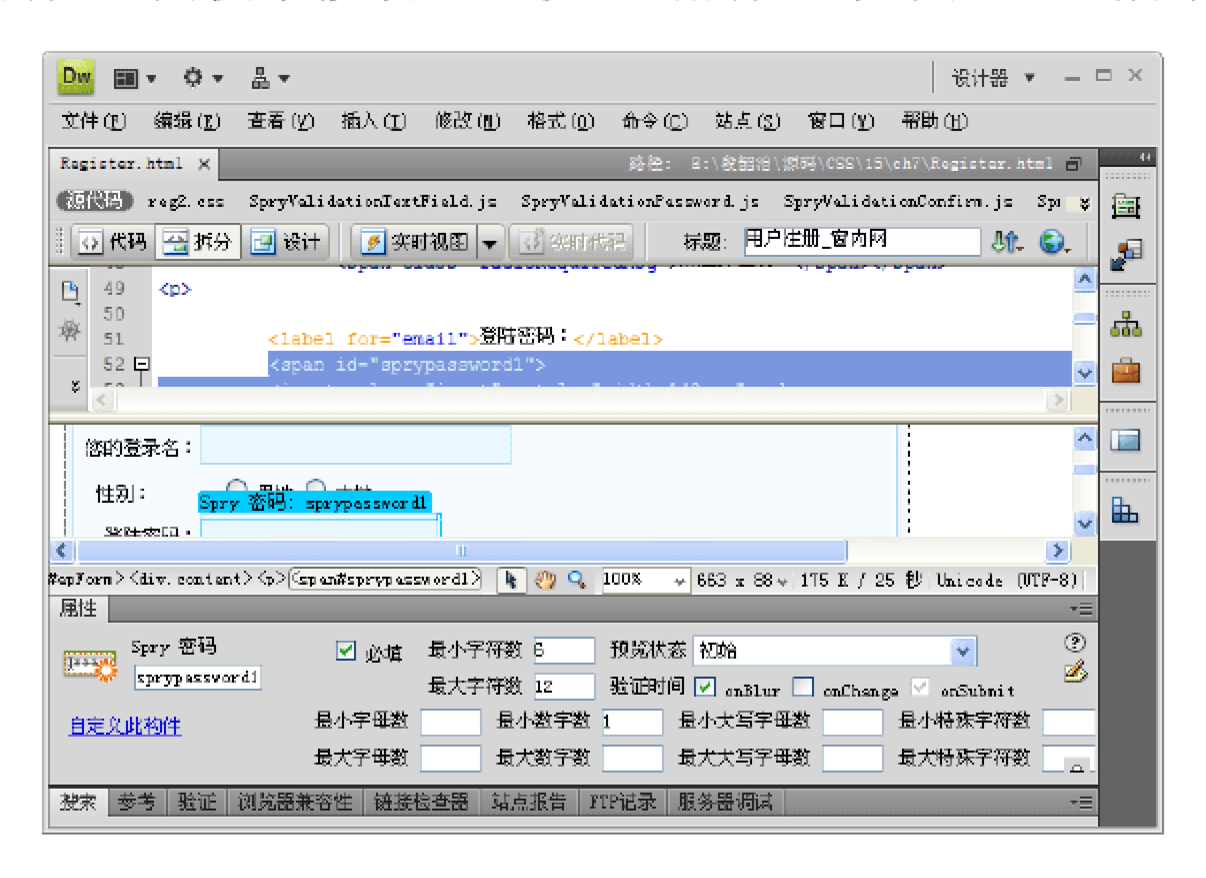


图 15-64　Spry 验证密码“属性”面板

Spry 验证密码组件可以在不同的时间点进行验证。例如，当用户在文本域外部单击时、输入内容时或尝试提交表单时。表 15-7 中对“属性”面板中的选项进行了解释。

在“属性”面板中启用 onBlur 复选框，设置允许用户在密码域中输入 6～12 位字符，其中至少应该包含 2 个字母和 1 个特殊字符。

表 15-7　Spry 文本区域属性参数

名　称		功　能
Spry 密码		标识密码的名称，该名称也是密码的 ID 号
预览状态	初始状态	在浏览器中加载页面或用户重置表单时组件的状态
	焦点状态	当用户将光标放置在组件上时的状态
	有效状态	当用户正确地输入信息且表单可以提交时组件的状态
	必填状态	用户必须在该文本区域内输入数据
	强度无效状态	当用户输入的文本不符合密码文本域的强度条件时
	最小字符数状态	当用户输入的字符数小于密码文本域中所需的最小字符数时
	最大字符数状态	当用户输入的字符数大于密码文本域中允许的最大字符数时
验证时间	onBlur	该项为“模糊”状态，即当用户在文本域的外部单击时验证
	onChange	该项为“更改”状态，即当用户更改文本域中的文本时验证
	onSubmit	该项为“提交”状态，即当用户尝试提交表单时验证
最小字母数		设置用户在密码域中最少应该输入的字母数量
最大字母数		设置允许用户在密码域中最多输入的字母数量
最小数字数		设置用户在密码域中最少应该输入的数字数量
最大数字数		设置允许用户在密码域中最多输入的数字数量
最小大写字母数		设置用户在密码域中最少应该输入的大写字母数量

续表

名　称	功　能
最大大写字母数	设置允许用户在密码域中最多输入的大写字母数量
最小特殊字符数	设置用户在密码域中最少应该输入除数字、字母之外的字符数量
最大特殊字符数	设置允许用户在密码域中最多输入除数字、字母之外的字符数量

现在，按 F12 键预览 Spry 验证密码组件的效果。默认运行后，当 “密码”文本框为空时会提示“需要输入一个值”，并以“红色”背景显示。此时，如果输入一些字符，在小于 6 位时(不满足最小字符数)就会提示“不符合最小字符数”。

在输入的字符数超过 12 位时(最大字符数)会提示“已超过最大字符数”，如图 15-65 所示。在允许的字符数内，如果密码域缺少字母或特殊字符都会提示“密码未达到指定的强度”，如图 15-66 所示。图 15-67 所示为正确输入时的密码域效果，没有任何提示，并且以“绿色”背景显示。

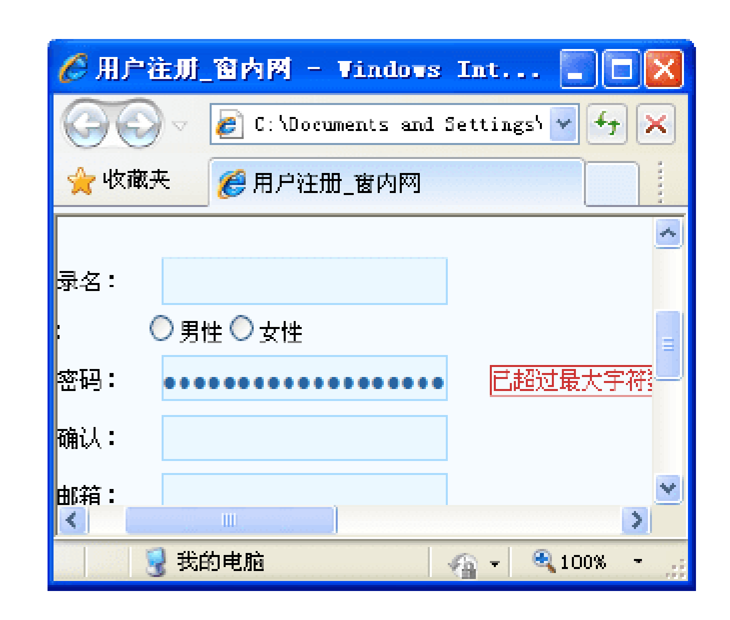


图 15-65　已超过最大字符数效果

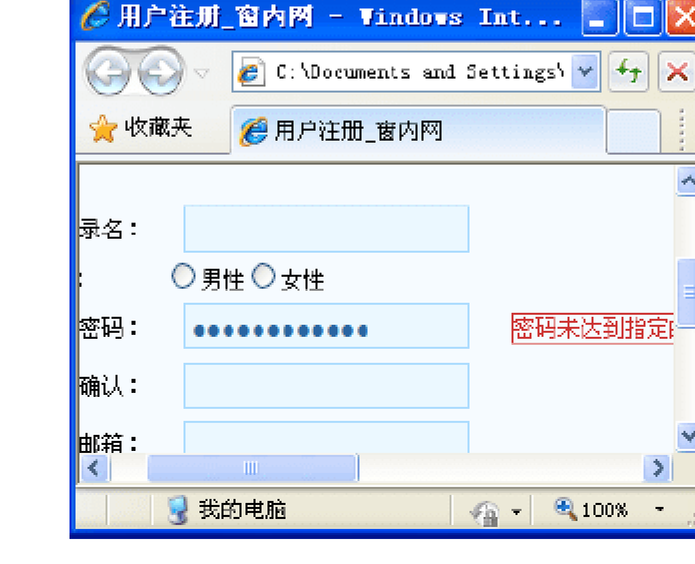


图 15-66　大于强度效果

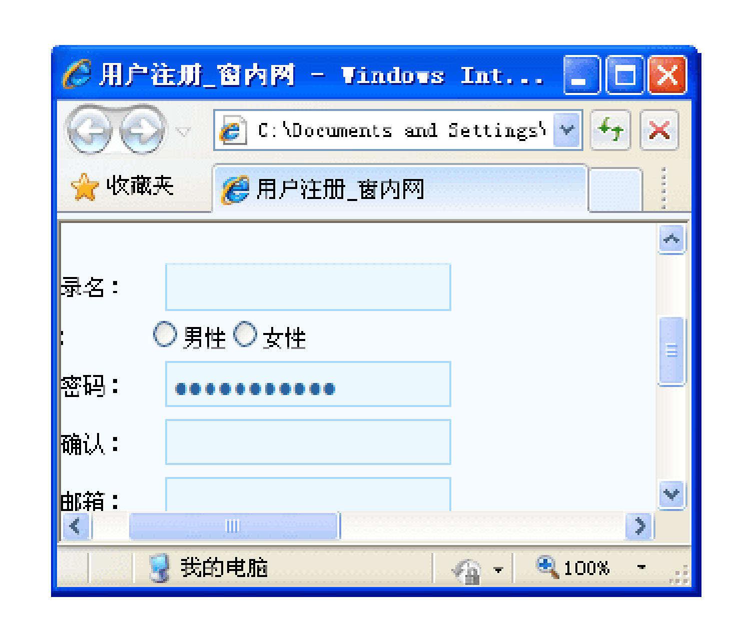


图 15-67　输入正确效果

Spry 验证确认组件是一个文本域或密码表单域。当用户输入的值与同一个表单中类似域的值不匹配时，该组件将显示有效或无效状态。例如，用户可以向表单中添加一个验证确认组件，要求用户重新输入在上一个域中指定的密码。如果用户未能完全一样地输入他们之前指定的密码，组件将返回错误信息，提示两个值不匹配。

在“表单”选项卡中，单击“Spry 验证确认”按钮 Spry 验证确认，弹出“输入标签辅助功能属性”对话框。并在“标签”文本框中输入“重复密码”，单击“确定”按钮，如图 15-68 所示。

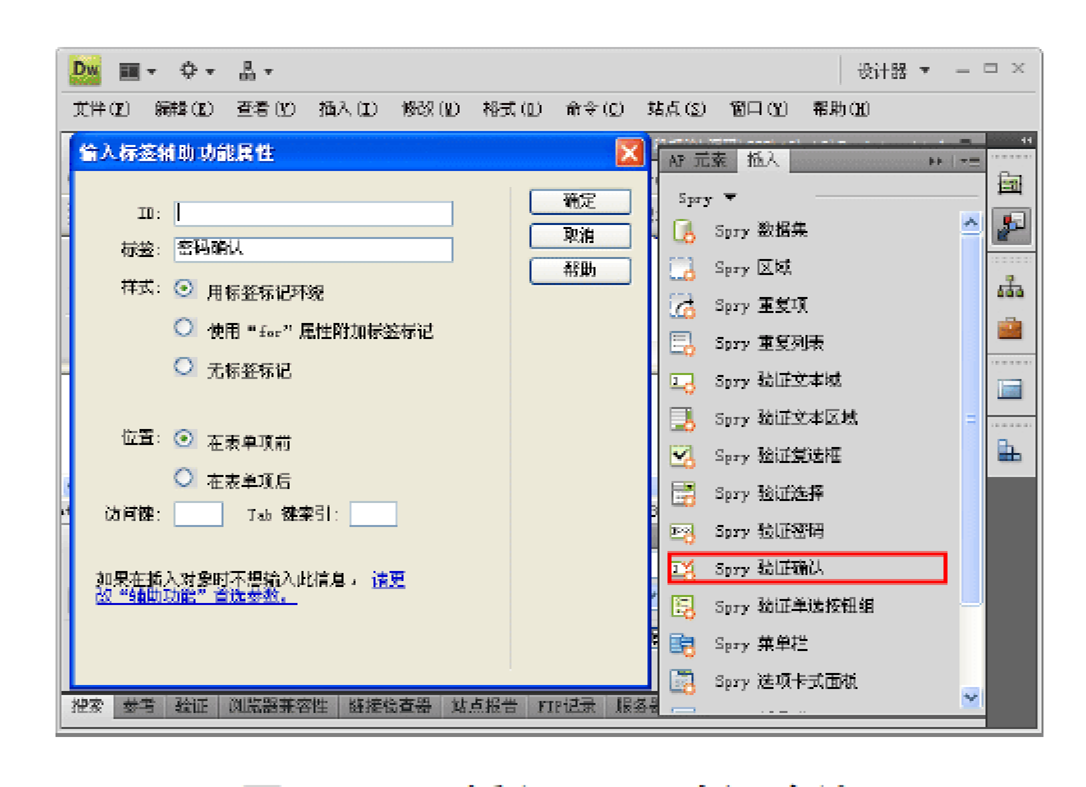


图 15-68　插入 Spry 验证确认

默认情况下，使用 Dreamweaver CS4 插入的所有 Spry 验证确认组件在发布到网页时，要求用户输入内容。但是，用户可以将填写确认文本域设置为是可选的。如图 15-69 所示为 Spry 验证确认组件的“属性”面板，在这里可以对组件所需的各种选项进行设置。

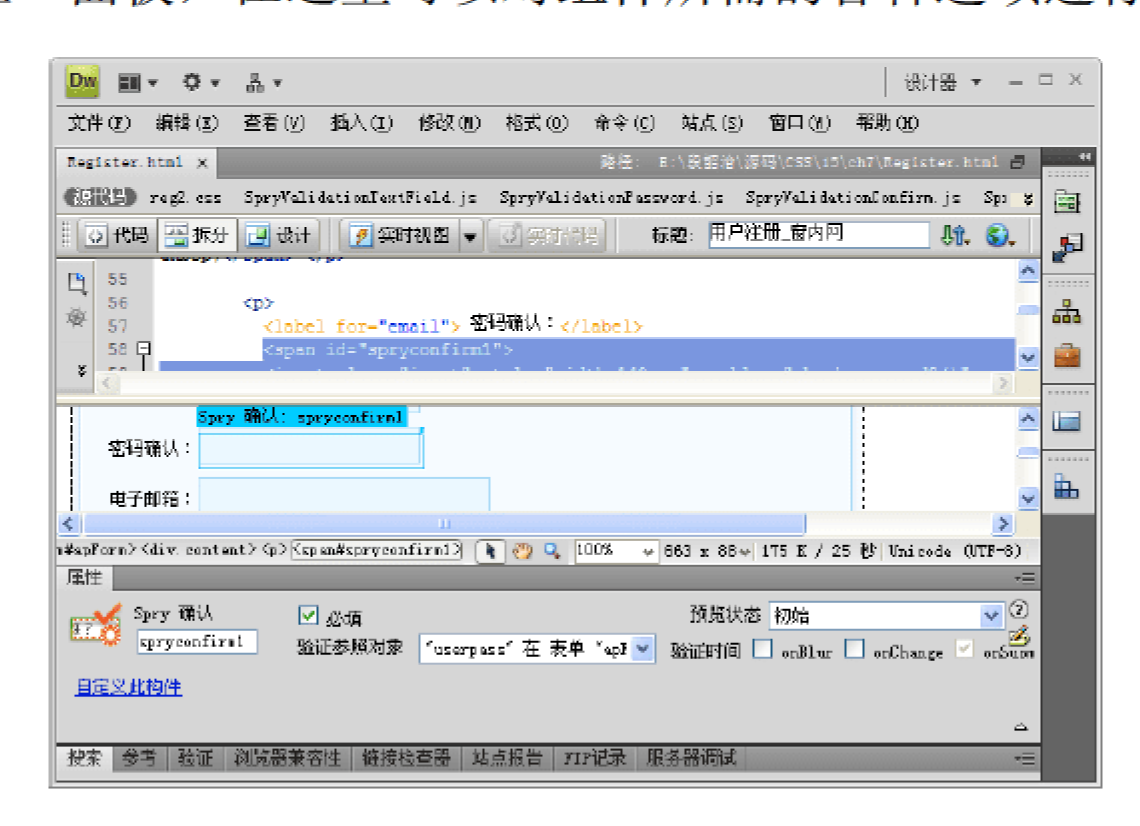


图 15-69　Spry 验证确认的“属性”面板

Spry 验证确认组件的属性非常少。除了其他组件共有的预览状态和验证时间外，仅包含一个“验证参照对象”属性。该属性用于选择用于验证依据的文本域，文本域可以是页面上分配了唯一 ID 的所有项，均可以在下拉列表框中选择。

这里选择 15.6 节使用 Spry 验证密码组件创建的“password1”，如图 15-70 所示。

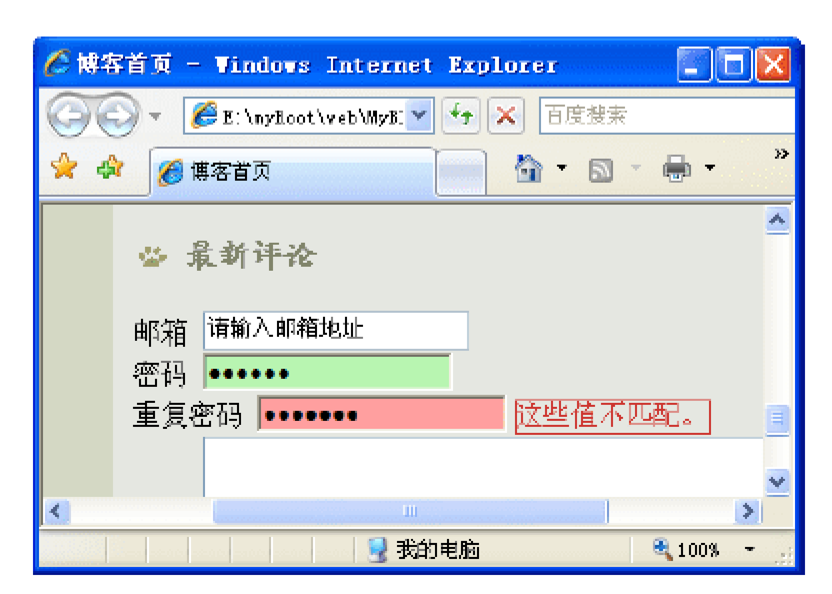




图 15-70　预览 Spry 验证确认效果



Spry 验证确认组件还可以与验证文本域组件一起使用，用于验证电子邮件地址。

Spry 验证单选按钮组件是一组单选按钮，可支持对所选内容进行验证，该组件可强制从组件中选择一个单选按钮。单击 Spry 验证单选按钮组 按钮则可以执行相关插入操作。

此时会弹出“Spry 验证单选按钮组”对话框，在“名称”文本框中输入该单选按钮组的名称。通过单击“添加”按钮 + 或者“删除”按钮 − 向组中添加或者从组中删除单选按钮。在“标签”列中，单击每个单选按钮的名称可以进行编辑，并为每个单选按钮分配唯一的名称。在“值”列中，单击对每个值进行编辑，并分配一个唯一的值。

在“布局方式”选项中默认选择了“换行符(
标签)”单选按钮，这种方式将会使用换行符(
标签)将每个单选按钮放置在单独的行中。如果选择“表”单选按钮则会使用单独的表格行(<tr>标签)将每个单选按钮放置在单独的行中。

最后，再单击【确定】按钮将 Spry 验证单选按钮组组件插入页面，如图 15-71 所示。

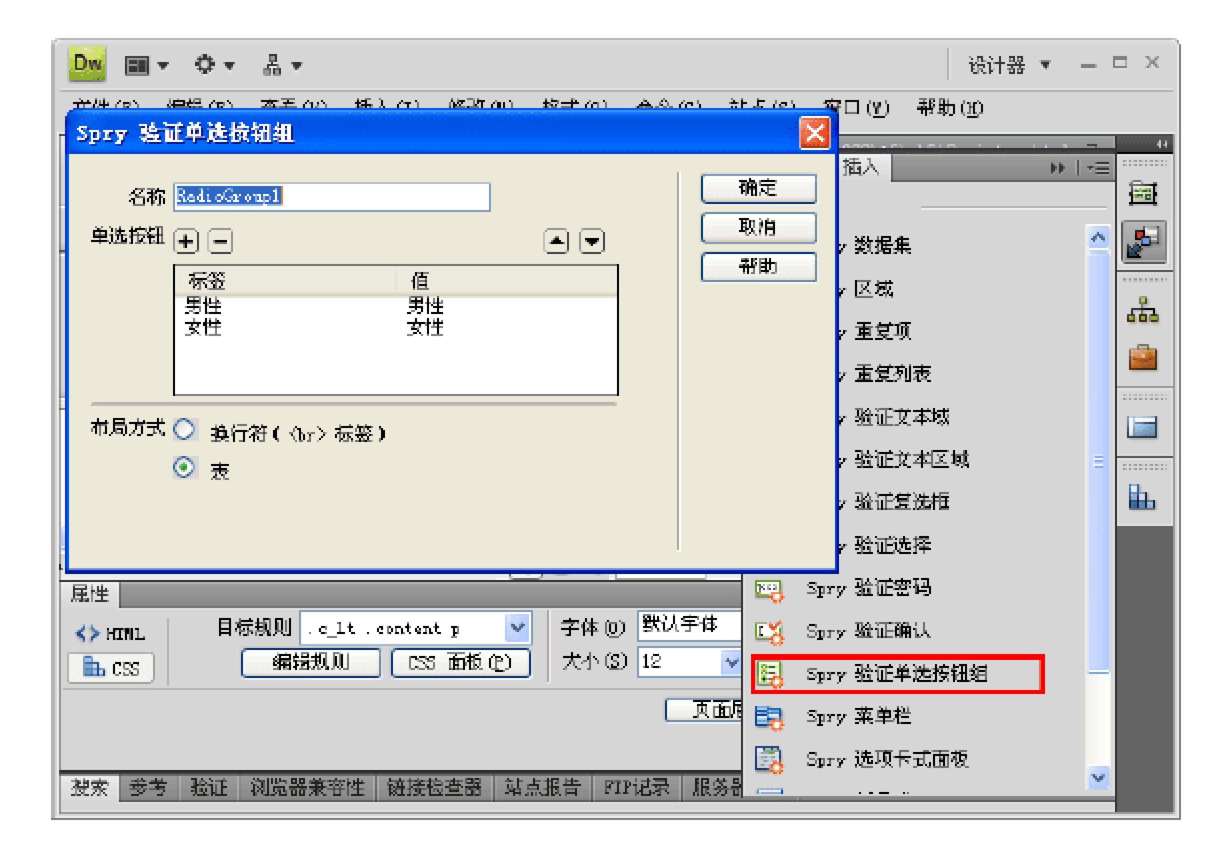


图 15-71　插入 Spry 验证单选按钮组

这里以默认的布局方式为例，为 Spry 验证单选按钮组添加适量标签并设置值。打开组件的“属性”面板并启用 onBlur 复选框，在“空值”文本框中输入“1”，在“无效值”文本框中输入“2”，如图 15-72 所示。

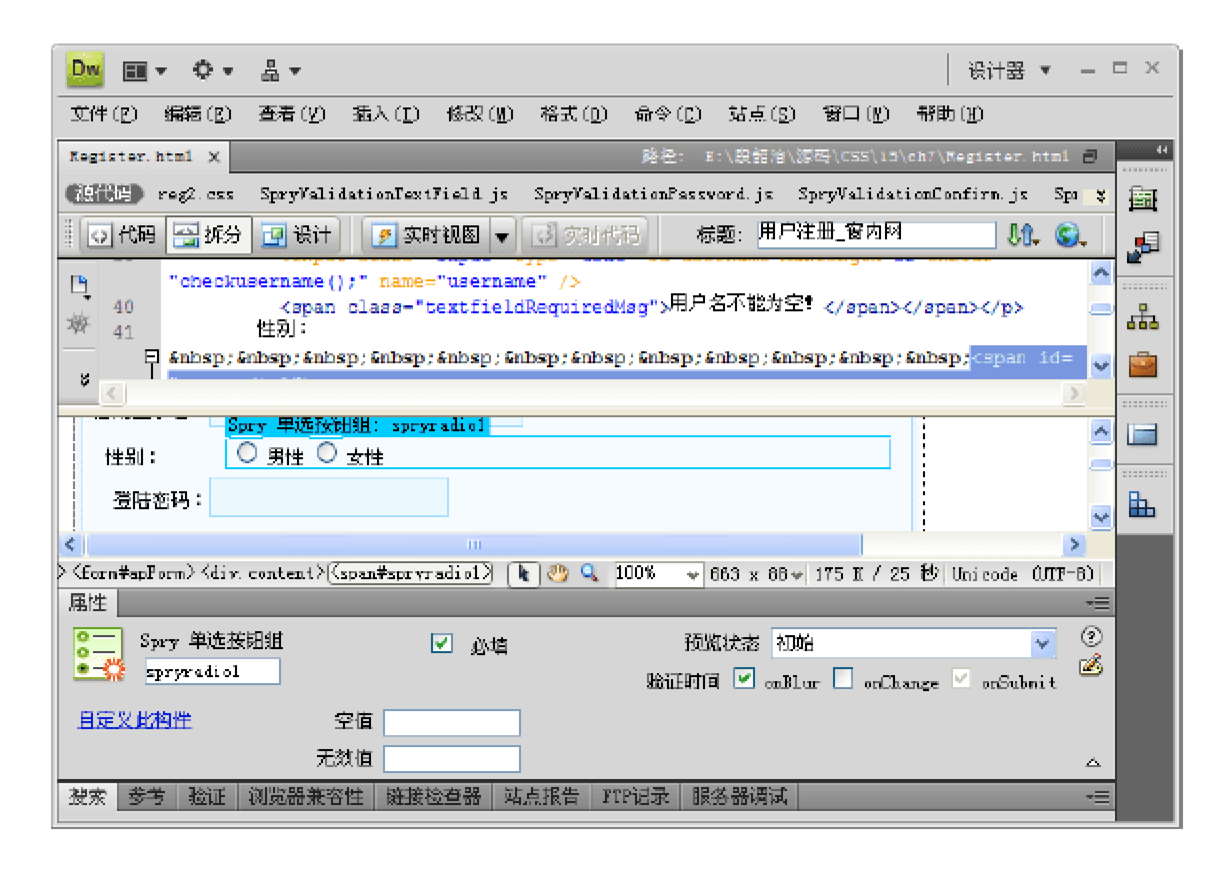


图 15-72　设置 Spry 验证单选按钮组属性

在本实例中值“1”对应“男性”单选按钮，值“2”对应“女性”单选按钮。这样如果用户选择具有空值的单选按钮，则浏览器将返回“请进行选择”错误消息，如图 15-73 所示。

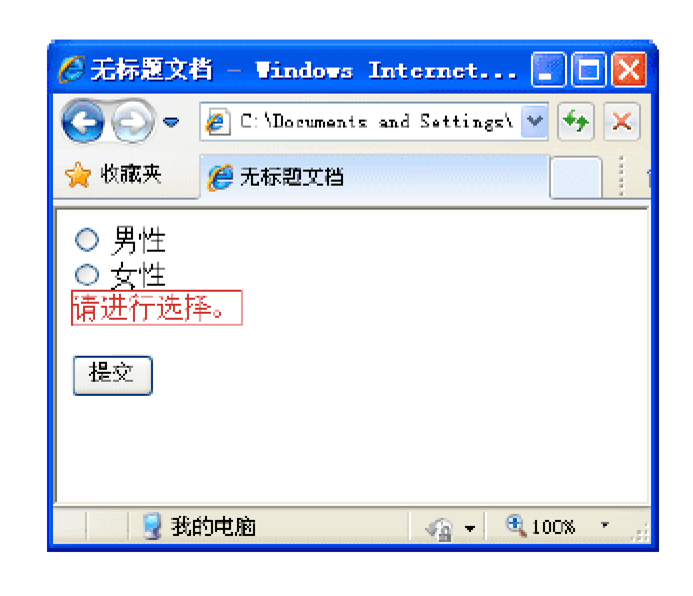


图 15-73　使用空值时

Spry 验证选择组件是一个下拉菜单，该菜单在用户选择时会显示组件的状态(有效或无效)。例如，插入一个包含状态列表的验证选择组件，这些状态按不同的部分组合并用水平线分隔。如果用户意外选择了某条分界线(而不是某个状态)，验证选择组件会向用户返回一条消息，声明选择无效。

在“表单”选项卡中，单击“Spry 验证选择”按钮 Spry 验证选择，弹出“输入标签辅助功能属性”对话框。并在“标签”文本框中输入“出生地”，单击“确定”按钮，如图 15-74 所示。

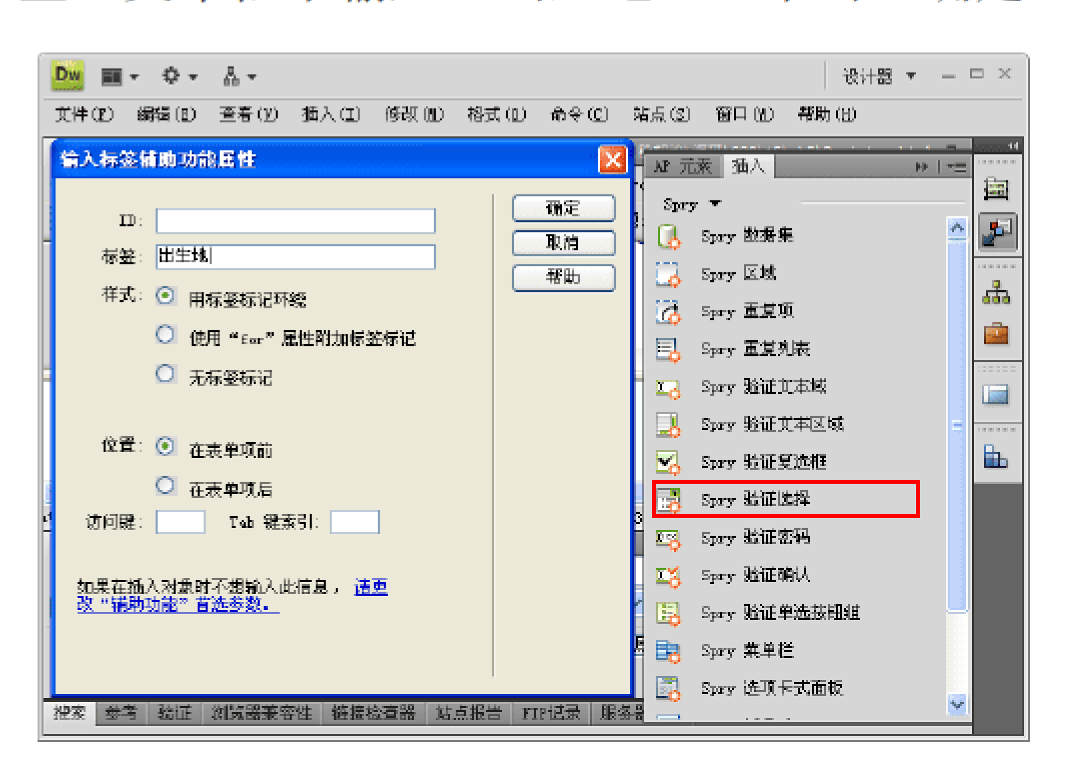


图 15-74　插入 Spry 验证选择



此时，将弹出 Spry 验证选择“属性”面板，并在该面板中设置验证选择是否为空值、预览状态，以及设置验证发生的时间。

现在，用户可以选择菜单域，单击“属性”面板中的“列表值”按钮。然后，在弹出的“列表值”对话框中，单击“添加”按钮，并添加“项目标签”和“值”内容，如表 15-8 所示。

表 15-8　添加列表值内容

项目标签	值
请选择地区	(空)
中国	-1
河南	1
山西	2
河北	3
湖北	4

最后，切换到“Spry 选择：spryselect1”选项卡，在“属性”面板中选中“空值”和“无效值”复选框，并设置“无效值”为-1，再设置“预览状态”为“初始”，选中 onBlur 复选框，如图 15-75 所示。

此时，按 F12 键预览 Spry 验证选择效果。例如，从“出生地”列表中的“请选择地区”项时，提示“请选择出生地”信息，并以“红色”背景显示；如果选择了出生地则以“绿色”背景显示，如图 15-76 所示。

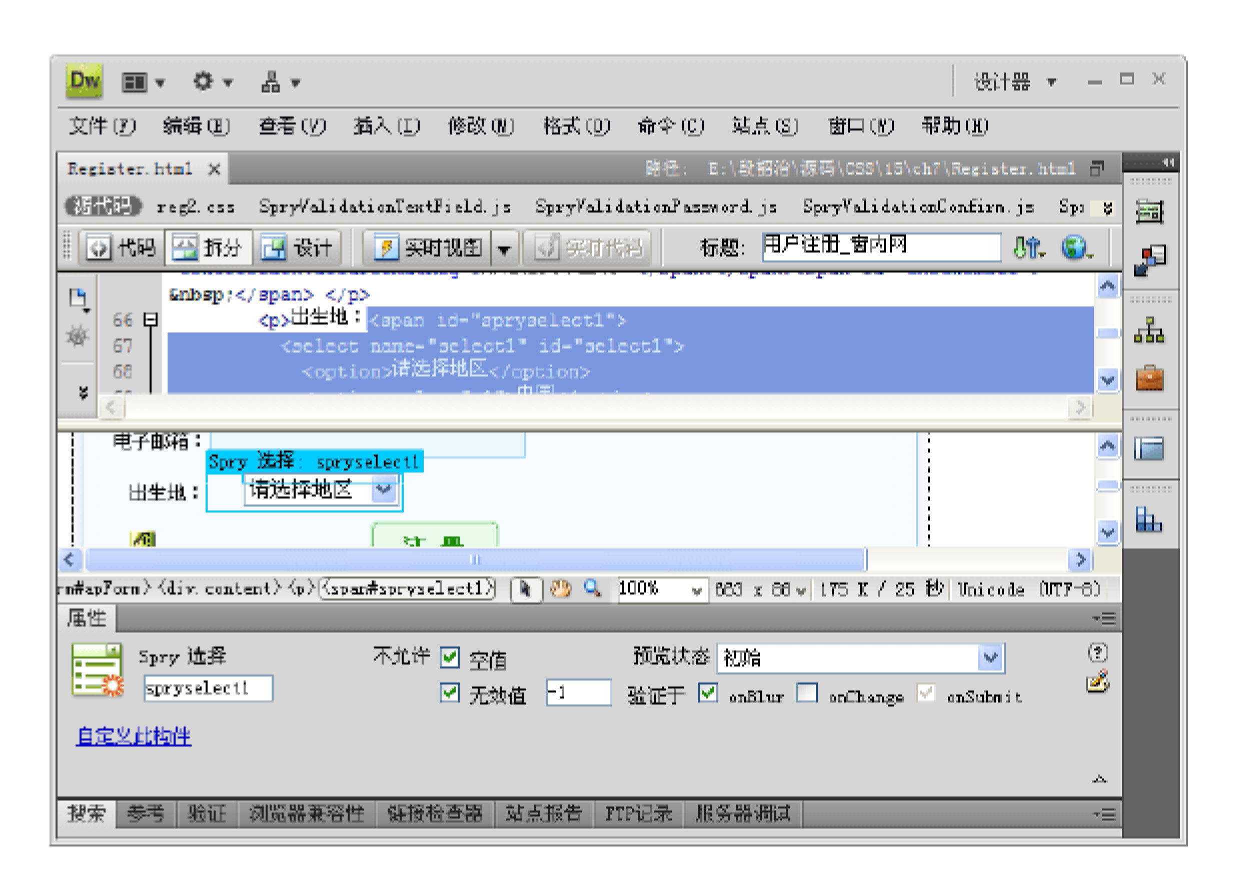


图 15-75　设置属性参数



当用户设置 Spry 验证文本区域“属性”面板后，还可以在“CSS 样式”面板中设置该验证文本区域 CSS 规则。其设置方法与 Spry 验证文本域相同。



图 15-76　预览 Spry 验证选择效果

15.7.3　运行结果

运行 Register.html 页面，运行结果如图 15-77 所示。



图 15-77　用户注册运行结果图

15.7.4　实例分析



源码解析：

本实例通过 Dreamweaver CS4 中 Spry 框架的内置组件制作了一个用户注册页面。这些组件主要有：文本域、密码域、密码确认、单选按钮和选择列表等。

通过使用 Spry 框架提供的丰富用户交互组件，使得客户端的开发和验证更加方便和快捷。

15.8　常见问题解答

15.8.1　将样式保存至本地服务器



如何将生成的样式保存至本地服务器上呢？

网络课堂：http://bbs.itzcn.com/thread-12313-1-1.html

今天使用 Dreamweaver CS4 生成了整个系统的 CSS 样式信息，可是发现页面代码太多了，而且 70%都是 CSS 样式表，我可否将样式信息保存至本地呢？

【解决办法】可以将 CSS 样式保存至本地服务器，只需要在“新建 CSS 规则”对话框下面的“规则定义”下拉列表框中选择“新建样式表文件”选项，如图 15-78 所示，默认选择“仅限该文档”选项。

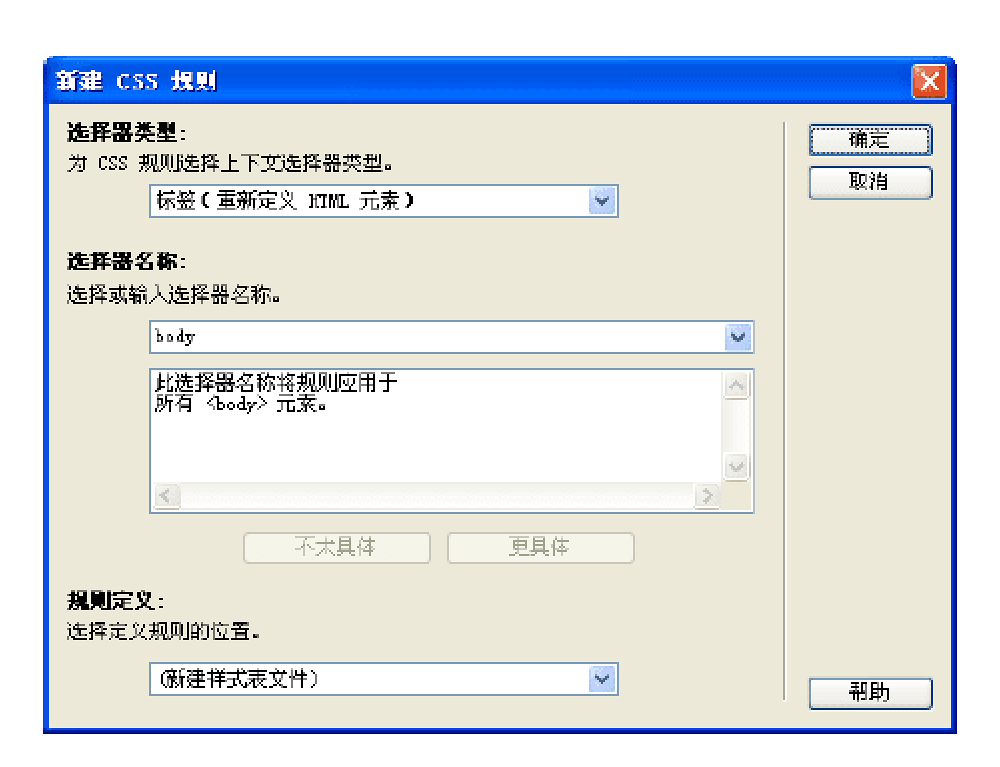


图 15-78　“新建 CSS 规则”对话框

15.8.2　Spry 框架设置超链接



使用 Spry 框架创建菜单栏时主菜单是否可以设置超链接？

网络课堂：http://bbs.itzcn.com/thread-12314-1-1.html

创建好菜单栏后，若希望在创建的主菜单选项上设置一个超链接，是否可以直接像子菜单一样设置，如图 15-79 所示。

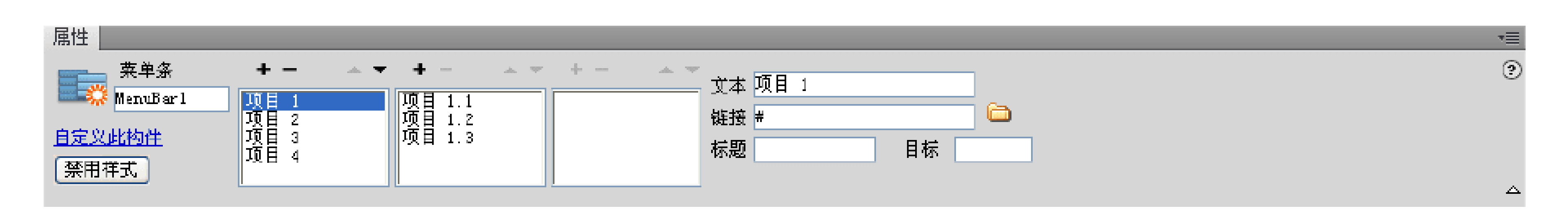


图 15-79　菜单栏属性

【解决办法】即使具有子菜单的菜单栏目也可以设置超级链接，这样既可以通过该菜单栏目显示相应的子菜单，也可以单击该菜单栏目跳转到其他网页。

15.9　习　　题

1. 填空题

(1) Dreamweaver 是一个所见即所得的编辑器，在 HTML 代码编辑方面提供了非常强大的功能。其中可以在__________、代码视图与拆分视图中分别进行查看与编辑。

(2) 编码工具栏包含可用于执行多种标准编码操作的按钮，例如折叠和展开所选代码、应用和删除注释、__________、插入最近使用过的代码片段等。

(3) Spry 框架支持一组用标准 HTML、CSS 和__________编写的可重用组件。

(4) Spry 框架中的每个组件都与唯一的 CSS 和__________文件相关联。CSS 文件中包含设置组件样式所需的全部信息，而 JavaScript 文件则赋予组件功能。

2. 选择题

(1) 在“新建 CSS 规则”对话框中没有的下拉列表框是__________。

A. “选择器类型”下拉列表框　　B. “选择器名称”下拉列表框

C. “菜单功能”下拉列表框　　D. “规则定义”下拉列表框

(2) Spry 组件是一个页面元素，通过启用用户交互来提供更丰富的用户体验。每个 Spry 组件均由几个部分组成，以下选项中属于该组件的部分是__________。

A. 组件名称　　B. 组件结构　　C. 组将动作　　D. 组件形成

(3) 使用 Spry 框架可以在页面中轻松制作出页面展示特效，下面不属于该框架的内容是__________。

A. 折叠菜单　　B. 多标签式选项卡

C. 垂直/水平多级导航　　D. 统计报表

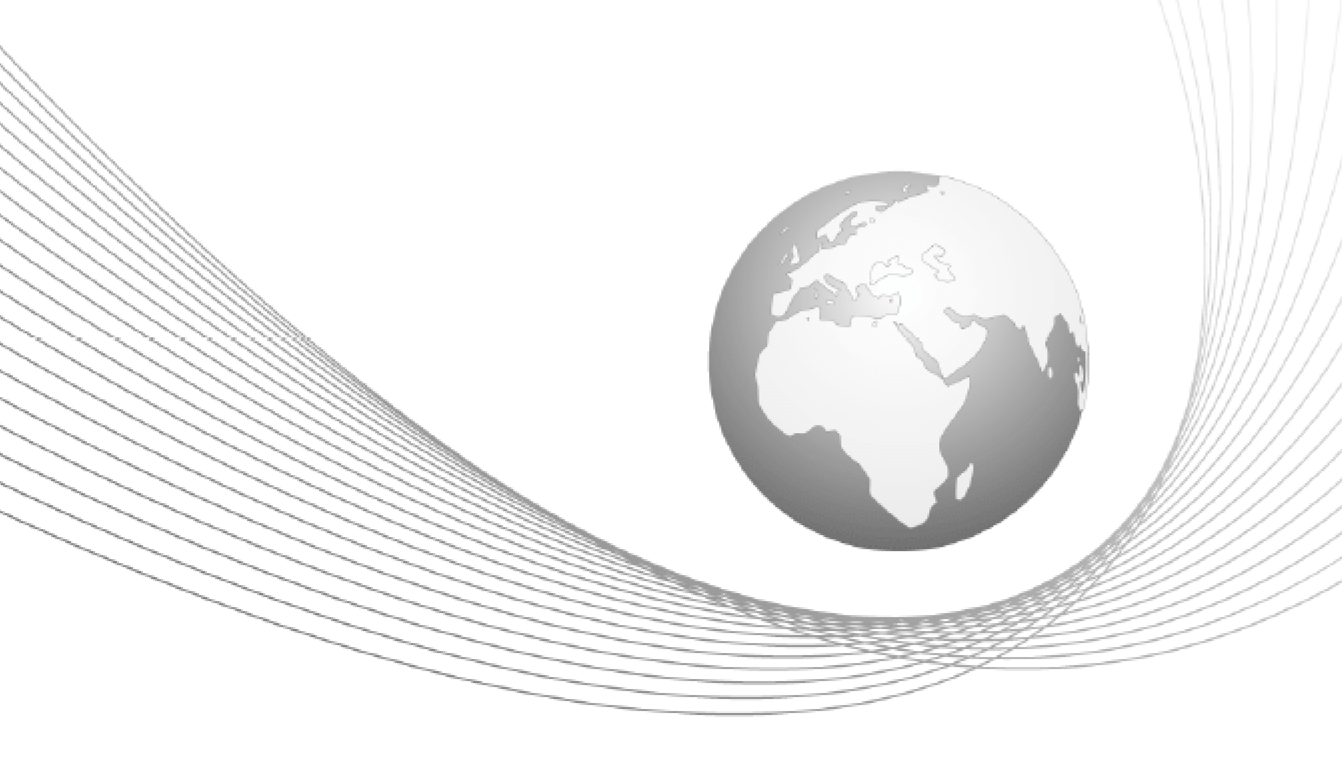
3. 上机练习

上机练习：最新日志推荐

实例要求：使用 Spry 框架中的选项卡面板制作出最新日志推荐、周单击量、月单击量和单击排行的栏目切换功能，运行结果如图 15-80 所示。



图 15-80　最新日志推荐



第 16 章　与 XML 整合

内容摘要：

XML 是一种描述性语言，用于对网络数据的转换和描述。与 HTML 不同，XML 没有规定的标记，只要符合 XML 规范，用户就可以根据需要自行创建标记来描述 XML 文档要表达的内容，并且可以在 XML 文档中调用 CSS 样式进行优化，这样可以制作出规范的网页。本章将介绍如何创建一个 XML 文档、如何在 XML 文档中调用 CSS 样式文件和 XML 文档，如何实现各行变色的表格。

学习目标：

- 掌握 XML 文档的基本语法
- 掌握如何在 XML 文档中调用 CSS 样式文件
- 掌握 XML 实现表格隔行变色的方法

16.1 创建简单的 XML 文件

XML(Extensible Markup Language，即可扩展标记语言)与 HTML 一样，都是 SGML(Standard Generalized Markup Language，标准通用标记语言)。XML 是 Internet 环境中跨平台的，依赖于内容的技术，是当前处理结构化文档信息的有力工具。扩展标记语言 XML 是一种简单的数据存储语言，其使用一系列简单的标记描述数据，而这些标记可以用方便的方式建立，虽然 XML 比二进制数据要占用更多的空间，但 XML 极其简单且易于掌握和使用。

XML 是一种元标记语言，它可以在多种领域应用，虽然 XML 允许用户自行创建标记，但是必须遵循 XML 规范及文档结构。本节将讲解如何创建一个 XML 文档。

**视频教学：**光盘/视频/16/16.1　创建简单的 XML 文件.avi　　长度：12 分钟

16.1.1 基础知识——创建 XML 文档

随着 Web 的广泛使用，HTML 的局限性逐渐暴露出来。最重要的就是 HTML 缺乏扩展性。为了满足 Web 的交互操作性和可扩展性的要求，W3C 推出了一种标准化的、具备高扩展性的、结构严谨的新语言——XML。

1. XML 简介

XML 具有很高的灵活性，但是必须遵循一定的结构，才能保证 XML 文档具有规范性。XML 文档结构可以分为三个部分：序言、主体和尾声。其中，序言和主体是每个 XML 文档必须保留的，而尾声则可以根据用户的需要来使用。

1) 序言

序言是 XML 文档的开始，用来表示对 XML 数据进行编译的开始及描述字符的编码方法，并为 XML 解析器和应用程序提供其他一些配置线索。序言部分格式代码如下：

```
<?xml version=“1.0” encoding=“编码” standalone=“yes/no”?>
```

格式中各参数的含义如下。

- version：指定了所采用的 XML 版本号，通常情况下，目前 Web 支持的是 XML 1.0 版本。
- encoding：指定了文档的编码格式。常用的编码格式有 UTF-8 和 GB2312 两种。在 XML 文档中未指定编码格式的情况下，默认为 UTF-8 格式。使用 UTF-8 格式，标记以及标记的内容就可以使用中文、日文、英文等，XML 解析器就会识别这些标记并正确解析标记中的内容，使用 GB2312 格式，则标记以及标记的内容只可以使用 ASCII 字符和中文。
- standalone：表示使用的 DTD 的形式，值只能是 yes 或 no，默认值为 no。如果为 yes，说明所有必需的实体声明都包含在文档中；如果是 no，则说明需要引用外部实体。

2) 主体

主体就是 XML 文档描述数据的地方，通常由标记、元素和属性等构件组成。

3) 尾声

XML 文档的尾声部分内容包括注释、处理指令和/或紧跟元素树后的空白。由于大多数应用程序在文档根元素的结束标记处就结束了，而不再对尾声进行任何处理，所以尾声部分对于 XML 文档来说不起任何作用。用户可以根据需要选择是否添加尾声。

2. XML 标记与元素

XML 是一种基于文本的标记语言。标记与元素在文档中是必不可少的，它们是构成 XML 文档最重要的构件。

1) 标记

在 XML 中，标记可分为非空标记和空标记两种。

非空标记由尖括号(<>)来界定，以左尖括号(<)开始，以右尖括号(>)结束。例如，给出一个 XML 文档，如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
    <学生>
        <姓名>李赞红</姓名>
        <性别>男</性别>
        <专业>计算机</专业>
    </学生>
```

在该 XML 文档中，<学生>和</学生>都是标记，<学生>称为开始标记，</学生>称为结束标记。同样，<性别>和<专业>都是开始标记，</性别>和</专业>则为结束标记。非空标记在开始标记和结束标记之间必须有数据的标记。



在 XML 文档中，开始标记和结束标记都是成对出现的。

空标记以“<”开始，以“/>”结束。空标记不标记任何内容，所以空标记没有开始标记和结束标记，但空标记内可以包含属性，如下所示：

```
<学生 />
<学生 姓名="李赞红" 性别="男" 专业="英语"/>
```

在第二个非空标记中，姓名、性别和专业都是标记学生包含的属性。



无论是非空标记还是空标记，在左尖括号“<”前不能有空格。

2) 元素

元素也是 XML 文档中最重要的构件。元素可以用来描述其所包含的数据，还可以包含属性名称和值，用于提供有关内容的其他信息，并指出这些信息的逻辑结构。在 XML 文档中，只有一个根元素，其他元素在根元素内是以树形分层结构来排列的，而且元素是可以嵌套使用的。

在 XML 中，元素可分为非空元素和空元素两种类型。

一个非空元素是由开始标记、结束标记及两标记之间的数据构成的。开始标记和结束标记用来描述标记之间的数据。标记之间的数据被认为是元素的值。非空元素的语法结构如下：

```
<开始标记>数据内容</结束标记>
```

而空元素就是不包含任何内容的元素，其语法结构如下：

```
<开始标记></结束标记>
```

在 XML 文档中，规范的 XML 标记与元素有助于正确地描述数据，所以它们必须遵循如下命名规则。

- 元素名中可以包含字母、数字和其他字符，如<place>、<地点>、<no123>等。元素名中虽然可以包含中文，但是有一些软件不能很好地支持这种命名，所以尽量使用英文字母来命名；也不要使用“:”，因为 XML 命名空间需要用到这个十分特殊的字符。
- 标记或元素名必须规范，名称必须以字母、下划线(_)或中文开头，而不能用数字或标点符号开头，中间不允许有空格，例如<book>、<_book>、<图书_book>等都是正确的名称。
- 名称中不能以字母 xml 的任意形式开头(如 XML 或 Xml 等)，如<xmlbook>、<Xmlbook>都是错误的。
- 名称中不能包含空格，如<图书 book>。
- 文档中的标记必须对应，每一个 XML 文档都必须有开始标记和结束标记。
- 标记区分大小写，例如标记<book>和<Book>是两个不同的标记。
- 元素标记必须合理地包含，在 XML 文档中不允许不合理的嵌套包含，即开始标记和结束标记要紧紧呼应。
- 对于元素中的开始标记和结束标记，要保证成对出现，并且大小写一致。

3. XML 属性

XML 属性可以将一些额外的信息附加到元素上，从而使文档对元素数据特性的描述更加具体。如果用户不希望通过子元素来描述元素的一些特性，则可以使用属性来存储。属性一般在元素的开始标记中声明，由属性名和值构成。它在非空元素和空元素中的语法格式如下：

```
非空元素 <标记名 属性列表>…</标记名>
空元素 <标记名 属性列表></标记名>
    或
    <标记名 属性列表/>
```

在 XML 中使用属性需要遵循下列规则。

- 属性名的命名规则和元素的命名规则相同，可以由字母、数字、中文及下划线组成，但必须以字母、中文或下划线开头，例如：sex、_sex 或性别都是正确的。
- 属性名区分大小写，例如：Sex 和 sex，虽然描述的都是姓名，但是编译时是两个不同的属性。
- 属性值必须使用单引号或双引号，例如："male"、'male'，它们描述的是相同的属性值。

● 如果属性值中要使用左尖括号“<”、右尖括号“>”、连接符号“&”、单引号“'”或双引号“"”时，必须使用实体引用。

在 XML 文档中，使用子元素和属性都可以实现对数据描述信息的存储。例如用元素和属性来描述各种桌子的尺寸，文档如下：

```
</xml version="1.0" encoding="UTF-8"?>
<桌子>
    <老板桌>
        <长>160cm</长>
        <宽>80cm</宽>
        <高>78cm</高>
    </老板桌>
    <书桌>
        <长>120cm</长>
        <宽>60cm</宽>
        <高>75cm</高>
    </书桌>
</桌>
```

XML 中，对于何时使用属性，何时使用子元素并没有特定的规则，只能根据执行环境和需要来决定使用哪个表现方式。但是，在文档中使用属性会引发下列问题。

● 属性不能包含多个数值(子元素可以)。
● 属性不容易扩展。
● 属性不能够描述文档结构(子元素可以)。
● 属性很难被程序代码处理。
● 属性值很难通过 DTD 进行测试。

建议读者尽量使用子元素而少用属性。由于 XML 的高扩展性，所以数据在 XML 中需要经常增加和修改。如果将增加或修改的数据放入属性中，那么将非常不利于对数据的维护和更新。

16.1.2　实例描述

我有一个朋友是画家，他的画室每天都会有各种各样的画展现给大家，每张画下面都会有一个简单的介绍，比如说这张画的名称、作者，以及作者的国家、公司，当然每张画的价钱是最关键的。他每天都会在一个木板上面刻上不同的字来介绍某张画。我看到后，很是不解，何必这样跟自己过不去呢，于是我帮他写了个 XML 文件，只要在页面中读取该 XML 文件，然后打印在纸上，贴在木板上就 OK 了。

下面就让我们来看看当时我给他写的 XML 文件吧！

16.1.3　实例应用

【例 16-1】创建简单的 XML 文件

编辑 16-1.xml 文件，内容如下所示：

```
<?xml version="1.0" encoding="utf-8"?>
<CATALOG>
    <CD>
        <TITLE>Empire Burlesque</TITLE>
        <ARTIST>Bob Dylan</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>Columbia</COMPANY>
        <PRICE>10.90</PRICE>
        <YEAR>1985</YEAR>
    </CD>
    <CD>
        <TITLE>Hide your heart</TITLE>
        <ARTIST>Bonnie Tyler</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>CBS Records</COMPANY>
        <PRICE>9.90</PRICE>
        <YEAR>1988</YEAR>
    </CD>
</CATALOG>
```

16.1.4 实例分析


源码解析：

在这个实例中，16-1.xml 文件使用元素和属性相结合的方式实现了对数据的描述信息，其中，<CD></CD>是 XML 元素，<TITLE></TITLE>、<ARTIST></ARTIST>等是 XML 属性，这样结合使用，就完成了描述数据的功能。

16.2 在 XML 中引用外部 CSS 样式文件制作家电条幅

XML 关心的是数据的结构，并能很好、方便地描述数据，但它不提供数据的显示功能。因此，浏览器不能直接显示 XML 文件中标记的文本内容，如果想让浏览器显示 XML 文件中标记的文本内容，则必须以某种方式告诉浏览器如何显示。W3C 为 XML 数据显示发布了两个建议规范：CSS(层叠样式表)和 XSL(可扩展样式语言)，本节重点讲述怎样使用 CSS 显示 XML 标记中的文本内容。


视频教学：光盘/视频/16/16.3 在 XML 中嵌入 CSS 样式制作家电条幅.avi 长度：7 分钟

16.2.1 基础知识——使用 CSS 显示 XML 中的数据

在 CSS 中，最重要的概念就是样式表。样式表是一组规则，通过这组规则告诉浏览器用

什么样式来显示文本，比如，告诉浏览器使用什么样的字体、颜色和页边距来显示文本。一个样式表的格式如下：

```
文本代表
{
    样式规则
}
```

对于 XML 文件，样式表中的“文本代表”可以是标记的名称。样式表中的“样式规则”是若干个用分号分隔的“属性名：属性值”，例如：

```
TITLE
{
    display:block;font-size:36px;font-weight:bold;
}
```



对于 XML 文件，样式表中的“文本代表”可以是标记的名称，也可以是标记的名称与该标记的 ID 属性值用“#”连接起来的字符串。当 XML 有许多标记具有相同的名字，需要使用不同的外观来显示它们的内容时，“文本代表”使用“标记名称”+“#”+“ID 属性值”就显得很方便了，这样我们可以为这些名字的标记指定不同的 ID 属性值。

其中，“TITLE”表示页面中“<TITLE>与</TITLE>”之间的内容要引用的样式；“display:block;”告知浏览器将“<TITLE>...</TITLE>”所标记的文本内容显示在一个“块区域”；“font-size:36px;”设置文本字体的大小为 36 像素；“font-weight:bold;”的作用是让文本加粗显示。

为了让 XML 使用层叠样式表，XML 文件必须使用操作下面的操作指令将当前的 XML 文件关联到某个层迭样式表。

```
<?xml-stylesheet href="样式表的 URL" type="text/css"?>
```

样式表的 URL 如果是一个文件的名字，该文件必须和 XML 文件在同一目录中，例如：

```
<?xml-stylesheet href="show.css" type="text/css"?>
```

样式表的 URL 如果是一个链接，该链接必须是有效的和可访问的，例如：

```
<?xml-stylesheet href="http://www.yahoo.com/show.css" type="text/css"?>
```

16.2.2　实例描述

最近手头终于宽裕了点，就想着买台电视机看看，到了家电商场一看，条幅上写着几种电视机的规格和简介。商人真是聪明啊，这样一来让顾客很清晰地看到自己想要的是哪种规格的电视机，而不用在这么大的商场里面转悠半天，说不定到头来也没有自己心中想要的那种规格，这样岂不是很浪费自己的宝贵时间吗？如果在商场的进口处，用一张条幅很醒目地贴出来，可以为顾客节省很多时间啊！

下面我们就使用 XML 文件引用外部 CSS 文件制作这个条幅吧！

16.2.3 实例应用

【例 16-2】在 XML 中引用外部 CSS 样式文件制作家电条幅

(1) 新建 16-2.xml 文件，内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<root>
    <TV>
        电视机
            <price>
                三种价格：
                    <item1>34 寸纯平 1388 元/台</item1>
                    <item2>34 寸数字高清 1899 元/台</item2>
                    <item3>34 寸液晶 5999 元/台</item3>
            </price>
            <date>生产日期：2010 年 12 月</date>
            <made>北京电视机厂生产</made>
    </TV>
</root>
```

(2) 在 CSS 文件夹下新建 show.css 样式文件，设置上面的 XML 文件内容的样式。

(3) 在 show.css 样式文件中编辑“TV”标签样式，设置 16-2.xml 文件中<TV>与</TV>之间内容的样式，代码如下：

```
TV
{
    background-image:url(../images/431654.jpg);
    display:block;
    font-size:28px;
    font-weight:bold;
    color:black;
    height:600px;
}
```

(4) 在 show.css 样式文件中编辑 price 标签样式，设置 16-2.xml 文件中<price>与</price>之间内容的样式，代码如下：

```
price
{
    display:list-item;
    list-style-type:decimal;
    margin-left:30px;
    font-size:18px;
    color:red;
}
```

(5) 在 show.css 样式文件中编辑“item1”标签样式，设置 16-2.xml 文件中<item1>与</item1>

之间内容的样式，代码如下：

```
item1
{
    display:list-item;
    list-style-type:square;
    margin-left:60px;
    font-size:18px;
    font-weight:bold;
    color:red;
}
```

(6) 在 show.css 样式文件中编辑"item2"标签样式，设置 16-2.xml 文件中<item2>与</item2>之间内容的样式，代码如下：

```
item2
{
    display:list-item;
    list-style-type:square;
    margin-left:60px;
    font-size:18px;
    font-weight:bold;
    color:red;
}
```

(7) 在 show.css 样式文件中编辑"item3"标签样式，设置 16-2.xml 文件中<item3>与</item3>之间内容的样式，代码如下：

```
item3
{
    display:list-item;
    list-style-type:square;
    margin-left:60px;
    font-size:18px;
    font-weight:bold;
    color:red;
}
```

(8) 在 show.css 样式文件中编辑 "date" 标签样式，设置 16-2.xml 文件中<date>与</date>之间内容的样式，代码如下：

```
date
{
    display:list-item;
    list-style-type:decimal;
    margin-left:30px;
    font-size:18px;
    color:#9900FF;
}
```

(9) 在 show.css 样式文件中编辑"made"标签样式，设置 16-2.xml 文件中<made>与</made>

之间内容的样式，代码如下：

```
made
{
    display:list-item;
    list-style-type:decimal;
    margin-left:30px;
    font-size:18px;
    color:#9900FF;
}
```

(10) 最后在 16-2.xml 文件中引入 show.css 文件。在 16-2.xml 文件中添加：

```
<?xml-stylesheet href="css/show.css" type="text/css"?>
```

16.2.4 运行结果

在浏览器中运行 16-2.xml 文件，页面中就会显示数据的内容，如图 16-1 所示。



图 16-1 使用 CSS 显示 XML 中的数据

16.2.5 实例分析



源码解析：

在这个实例的 show.css 样式文件中，指定了“TV”标记的样式“height:600px;”，这样就指定了 XML 文件数据的显示区域高为 600 像素；同时还指定了“background-image:url(../images/431654.jpg);”，表示 XML 文件数据显示区域的背景图片为“../images/431654.jpg”，从而在浏览器中浏览 16-2.xml 文件时，出现了如图 16-1 所示的页面效果。

16.3　在 XML 中嵌入 CSS 样式制作家电条幅

在 XML 文档中引入 CSS 样式表有两种方法，一种是在 XML 文档中引入外部的 CSS 样式(前面一节中讲到的)；另一种是在 XML 文档中直接嵌入 CSS 样式。这节将为大家讲解一下如何在 XML 文档中嵌入 CSS 样式。

**视频教学：**光盘/视频/16/16.3　在 XML 中嵌入 CSS 样式制作家电条幅.avi　长度：7 分钟

16.3.1　基础知识——在 XML 中嵌入 CSS 样式

在 XML 中嵌入 CSS 样式也叫内嵌式，就是将 CSS 样式直接嵌入 XML 文档内部，为元素设置 style 属性，并在属性值中给出对其样式的定义。这种用法主要出现在一些特殊的 XML 文档中，一般来讲内嵌 CSS 样式的 XML 文档本身就是用于显示数据的，如 SVG、SMIL 等。

XML 文档如同内部的 DTD 语法，直接将 CSS 语法插入文件中，这并非标准的用法，事实上样式是使用 XHTML 标记<STYLE>在 XML 文档中插入 CSS 样式，语法如下：

```
<HTML:STYLE xmlns:HTML="http://www.w3.org/1999/xhtml">
……
</HTML:STYLE>
```

上述代码集声明了 XHTML 的命名空间代码 HTML，CSS 样式表定义在<HTML:STYLE>与</HTML:STYLE>标记之间，这样就完成了在 XML 文档中嵌入 CSS 样式，从而实现了在浏览器中直接显示 XML 文件的文本内容。

16.3.2　实例描述

16.2 节中，我们在编辑好的 XML 文档中引用外部的 CSS 样式来完成家电商场的条幅展示，当时，我们是通过编辑一个 XML 文件和一个 CSS 样式文件才完成了在浏览器中直接显示 XML 文件的文本内容的功能。其实，面对此类比较简单的数据展示，不必“大费周折”，直接在 XML 文档中嵌入 CSS 样式就可以了。

下面就一起来改编一下家电商场的条幅展示功能吧！

16.3.3　实例应用

【例 16-3】在 XML 中嵌入 CSS 样式制作家电条幅

编辑 16-3.xml 文件，在 XML 文档中嵌入 CSS 样式，内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/css"?>
<root>
    <HTML:STYLE xmlns:HTML="http://www.w3.org/1999/xhtml">
```

```
    /*和前面讲解的使用外部 css 样式文件制作家电条幅实例中的 css 样式一样，这里不再重复*/
    </HTML:STYLE>
    <TV>
        电视机
            <price>
                三种价格：
                    <item1>34 寸纯平 1388 元/台</item1>
                    <item2>34 寸数字高清 1899 元/台</item2>
                    <item3>34 寸液晶 5999 元/台</item3>
            </price>
            <date>生产日期：2010 年 12 月</date>
            <made>北京电视机厂生产</made>
    </TV>
</root>
```

上面代码中的第二行代码是 CSS 样式的声明，其中，xml-stylesheet 的意思是为 XML 文档添加样式表，type= "text/css"的意思是样式表的类型是 CSS 样式表。

代码中的<HTML:STYLE>是 CSS 样式内容的开始标记，<HTML:STYLE>与</HTML:STYLE>之间就是 CSS 样式内容。

16.3.4 运行结果

在浏览器中运行 16-3.xml 文件，出现如图 16-2 所示的界面。



图 16-2 在 XML 文件中嵌入 CSS 样式

16.3.5 实例分析



源码解析：

细心的同学不难看出，在这个实例中，我们只是把前面编辑的 show.css 样式内容复制到了<HTML:STYLE>与</HTML:STYLE>标记之间了，出现的界面效果是一样的。特别需要注意的是 XML 文件中的这行代码：<?xml-stylesheet type="text/css"?>是不可缺少的。

16.4　实现学生信息的隔行变色功能

也许你曾经利用 HTML 的<table>数据表格与 CSS 配合，制作出了方便使用的隔行变色表格。对于用 XML 表示的数据，同样可以采用类似的方法创建 XML 隔行变色表格。这节将介绍如何使 XML 文档中的数据表格实现隔行变色。

**视频教学**：光盘/视频/16/16.4　实现学生信息的隔行变色功能.avi　　长度：7 分钟

16.4.1　基础知识——XML 实现隔行变色的表格

前面我们提到过，对于 XML 文件，样式表中的“文本代表”可以是标记的名称，也可以是标记的名称与该标记的 ID 属性值用“#”连接起来的字符串，即“文本代表”使用“标记名称”+“#”+“ID 属性值”，那么如果需要使 XML 文档数据内容显示为隔行变色的表格时，使用“标记名称”+“#”+“ID 属性值”就显得很有价值了。我们完全可以为相同的标记指定不同的 ID 属性值来完成隔行变色的表格功能。

实现 XML 隔行变色表格需要先建立 XML 数据表格，它与 HTML 中的<table>不一样，通常需要对不同类型的单元格采用不同的标记，如下面的 XML 文档内容：

```
<title>
    <name>姓名</name>
    <class>班级</class>
    <birth>出生日期</birth>
    <constell>星座</constell>
    <mobile>固定电话</mobile>
</title>
```

上面定义了表格的列，即表格的标题行，接着就该为表格添加数据了。在编辑表格数据的 XML 时，指定偶数行的 ID 属性值，并且属性值是相同的，如下面的代码所示：

```
<student>
    <!--指定第一个学生的姓名、班级、出生日期、星座、固定电话属性值-->
</student>
<student id="stu">
    <!--指定第二个学生的姓名、班级、出生日期、星座、固定电话属性值-->
</student>
```

在上面代码中，指定了偶数行的<student>标记的 ID 属性值为“stu”，这样，可以使用“标记名称”+“#”+“ID 属性值”来命名样式名称，并指定该样式名称的样式规则，从而达到隔行换色的效果，如下面的 CSS 样式：

```
student#stu {
    background-color:#FFCCFF;       /*隔行变色 */
}
```

16.4.2 实例描述

最近在做一个学生管理系统，需要把学生的详细信息展现在表格中。这对于我们这些程序员来说是最简单不过的啦，但是，不要高兴得太早，我们需要的学生数据并非是从数据库中直接读取出来展现在页面中的，而是需要先生成一个 XML 文档，然后再展现。这难倒了我们公司的一大帮人，不过，程序员是有一定的“魄力”的，不会让这点小问题难倒。经过深思熟虑，终于想出了办法：直接在 XML 文档中引用 CSS 样式就可以实现表格的特效，同时，我们还可以让表格更加美观——隔行变色。下面就来做一下吧！

16.4.3 实例应用

【例 16-4】实现学生信息的隔行变色功能

(1) 创建 16-4.xml 文件，定义表格文件，内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="css/table.css" type="text/css"?>
<list>
    <!-- 标题行数据-->
    <title>
        <!-- 定义列名为"姓名"-->
        <name>姓名</name>
        <!-- 定义列名为"班级"-->
        <class> 班级</class>
        <!-- 定义列名为"出生日期"-->
        <birth> 出生日期</birth>
        <!-- 定义列名为"星座"-->
        <constell>星座</constell>
        <!-- 定义列名为"联系电话"-->
        <mobile>联系电话</mobile>
    </title>
    <!-- 设置表格的第一行数据，不变色-->
    <student>
        <!-- 指定"姓名"列的值-->
        <name>王丽丽</name>
        <!-- 指定"班级"列的值-->
        <class> 2 班</class>
        <!-- 指定"出生日期"列的值-->
        <birth> 1988-02-21</birth>
        <!-- 指定"星座"列的值-->
        <constell>双鱼座</constell>
        <!-- 指定"联系电话"列的值-->
        <mobile>12568487895</mobile>
    </student>
    <!-- 设置表格的第二行数据，变色-->
```

```
    <student id="stu">
        <!-- 指定"姓名"列的值-->
        <name>张赞红</name>
        <!-- 指定"班级"列的值-->
        <class>2 班</class>
        <!-- 指定"出生日期"列的值-->
        <birth>1988-01-24</birth>
        <!-- 指定"星座"列的值-->
        <constell> 双鱼座</constell>
        <!-- 指定"联系电话"列的值-->
        <mobile> 13658478957</mobile>
    </student>
    <!-- 设置表格的第三行数据，不变色-->
    <student>
        <!--数据和上面的两行数据相同，这里不再叙述-->
    </student>
    <!-- 设置表格的第四行数据，变色-->
    <student id="stu">
        <!--数据和上面的两行数据相同，这里不再叙述-->
    </student>
    <!-- 设置表格的第五行数据，变色-->
    <student>
        <name>杨芳</name>
        <class> 1 班</class>
        <birth>1988-06-07</birth>
        <constell> 双子座</constell>
        <mobile>15478487598</mobile>
    </student>
</list>
```

在 16-4.xml 文件中，定义的表格为 5 列，标题分别为 Name、Class、Birthday、Constellation 和 Mobile，并定义了表格数据为 4 行，同时定义了第 2 行和第 4 行数据的 ID 属性值为“stu”。

(2) 新建 CSS 样式文件 table.css，设置 16-4.xml 文件内容样式。

(3) 在 table.css 样式文件中定义“list”标签样式，设置 16-4.xml 文件中<list>与</list>之间的内容样式，代码如下：

```
list {
    font-family:Arial;
    Font-size:14px;
    line-height:25px;
    Position:absolute;      /*绝对定位 */
    top:-15px;
    left:0px;
    width:800px;
    height:600px;
}
```

(4) 在 table.css 样式文件中定义“title”标签样式，设置 16-4.xml 文件中<title>与</title>之间内容的样式，代码如下：

```
title {
    background-image:url(../images/title_bg2.jpg);
    display:block;                  /*块元素 */
    border:1px solid #0058a3;       /* 边框 */
    margin:20px 0px 0px 0px;        /*解决边框重叠的问题 */
    padding:0px 0px 0px 20px;
    font-size:18px;
}
```

(5) 在 table.css 样式文件中定义“student”标签样式，设置 16-4.xml 文件中<student>与</student>之间内容的样式，代码如下：

```
student {
    display:block;                  /*块元素 */
    background-color:#99CCFF;       /*背景色 */
    border:1px solid #0058a3;       /* 边框 */
    margin-bottom:- 1px;            /* 解决边框重叠的问题 */
}
```

(6) 使用“标记名称”+“#”+“ID 属性值”实现隔行变色功能，即在 table.css 样式文件中定义“student#stu”样式内容，设置 16-4.xml 文件中 id 为“stu”的<student>与</student>之间内容的样式，代码如下：

```
student#stu {
    background-color:#0099FF;/*隔行变色 */
}
```

(7) 设置表格标题样式，代码如下：

```
title name, title class, title birth, title constell, title mobile{
    color:#FFFFFF;/*行名称颜色 */
    font-weight:bold;
    text-align:center;
    padding:0 0 0 40;
}
```

(8) 设置表格内容样式，代码如下：

```
student name,student class, student birth,student constell, student mobile {
    padding:0 0 0 50;
    text-align:center;
}
```

(9) 编辑框架的顶部页面 header.htm 和左边导航菜单页面 menu.htm。

(10) 创建 index.htm 页面，将 header.htm 页面、menu.htm 页面和 16-4.xml 页面使用框架整合。

16.4.4 运行结果

使用浏览器运行 index.htm 文件，在页面中输出一个表格，如图 16-3 所示。



图 16-3　XML 实现隔行变色的表格

16.4.5　实例分析



源码解析：

在这个实例中，16-4.xml 文件指定了偶数行的<student>标记的 ID 属性值为“stu”，因此可以在 table.css 样式文件中编辑“student#stu”的样式规则，指定背景色，达到了换色的效果。

16.5　使用 XSLT 制作宠物展示页面

通过前几节的学习，我们已经知道，XML 注重的是数据的结构，并能很好且方便地描述数据。但是，数据的显示外观也是很重要的一个方面，XML 能有效地分离数据结构和数据显示，前面提到过，W3C 为 XML 数据显示发布了两个建议规范：CSS(层叠样式表)和 XSL(可扩展样式语言)，那么，这节将为大家讲述如何使用 XSL 显示标记中的文本内容。



视频教学：光盘/视频/16/16.5　使用 XSLT 制作宠物展示页面.avi　　长度：6 分钟

16.5.1　基础知识——使用 XSLT 显示 XML 数据

XSL(eXtensible Stylesheet Language)主要用来实现两个作用：数据表达和数据转换，但是，数据转换部分逐渐地从中独立出来形成了一个新的标准，即 XSLT(eXtensible Stylesheet Language Transformation)，而数据表达部分则形成了 XSL-FO，其中，XSLT 是 XSL 标准中最重要的一个部分。

1. XSLT 简介

XSLT 是为 XML 的样式显示设计的语言，该语言定义一系列元素集的 XML 的语法规范，

该语言规定可以把 XML 文档转换成 HTML、XML 或其他的文档，也就是说将一个 XML 文档转换成浏览器所能识别的一种格式。一个 XSLT 的样式表单中包含的是多个设计规则、显示方式，从 XML 文档中提取出来的数据依据这个样式表单规定的显示方式显示出来，形成一个新的文档。

XSLT 最常用的转换是将 XML 文档转换为 HTML 文档。XSLT 将每个 XML 元素都转换成一个 HTML 元素。XSLT 还可以向输出文件中增加全新的元素，或去掉一些元素。它可以重新安排这些元素并对元素进行分类，测试并确定显示哪些元素等。

相对于 CSS 只支持英文的限制，XSLT 最大的优点就是文档可以支持中文，这样可以更加适应用户的编写习惯。而且，目前的浏览器版本带有 XSLT 引擎的 XML 解析器，所以，完全具备了对 XSLT 文档的支持；用户可以直接使用 Internet Explorer 5.0 或者以上版本来调试 XSLT 文档。

2. XSLT 转换

一个能将 XSLT 文件转换为一个 HTML 文件的程序一定具有处理 XSL 样式表的功能，比如 IE5.5 就是具有这样功能的应用程序(IE5.5 带有 XSL 处理器)。当用浏览器打开一个和 XSLT 样式表关联的 XML 文件时，浏览器和 XML 关联的 XSLT 样式表转换为一个 HTML，并执行该 HTML 将数据显示在浏览器中。XSLT 转换的基本步骤如图 16-4 所示。

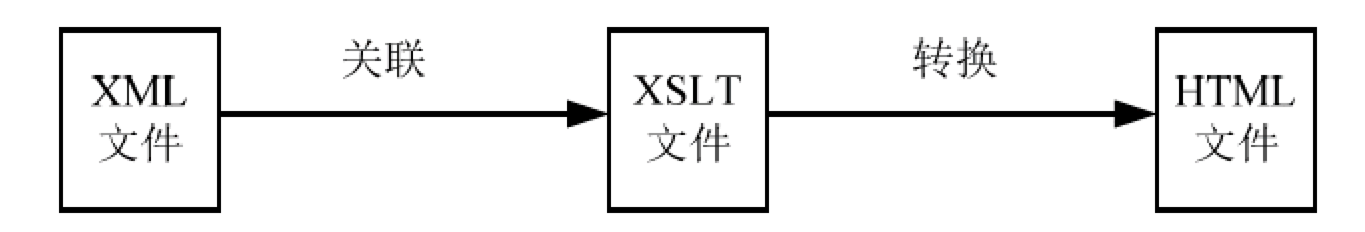


图 16-4　XSLT 转换的基本步骤

XML 文件需要使用类似下面的操作指令将 XML 文件和 XSLT 样式表关联：

```
<?xml-stylesheet href="xsl 样式表的 URL" type="text/xsl"?>
```

样式表的 URL 如果是一个文件的名字，该文件必须和 XML 文件在同一目录中，例如：

```
<?xml-stylesheet href="show.xsl" type="text/xsl"?>
```

样式表的 URL 如果是一个链接地址，该链接地址必须是有效的，例如：

```
<?xml-stylesheet href="www.itzcn.com/show.xsl" type="text/xsl"?>
```

讲述了如何在 XML 文档中引用外部的 XSLT 文件来显示页面。下面就按照上面的步骤做一个简单的例子吧！

创建 16-5.xml 文件，内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<名单>
    <姓名>
        张赞红
    </姓名>
    <出生日期>
        1989 年 02 月 21 日
    </出生日期>
</名单>
```

假设 XML 文件 16-5.xml 需要把标记的数据分别用黑体 1 和黑体 2 显示在浏览器中，那么需要创建 XSLT 文件，内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="GBK"/>
<xsl:template match="/">
    <HTML>
        H1>
            sl:value-of select="名单/姓名"/>
        </H1>
        <H2>
            <xsl:value-of select="名单/出生日期"/>
        </H2>
    <TML>
</xsl:template>
</xsl:stylesheet>
```

除了需要创建上面的 XSLT 文件之外，还需要在 XML 文档中关联 XSLT 文件。因此，需要在 16-5.xml 文件中添加下面的关联代码：

```
<?xml-stylesheet href="16-5.xsl" type="text/xsl"?>
```

用浏览器打开“16-5.xml”文件的效果如图 16-5 所示。

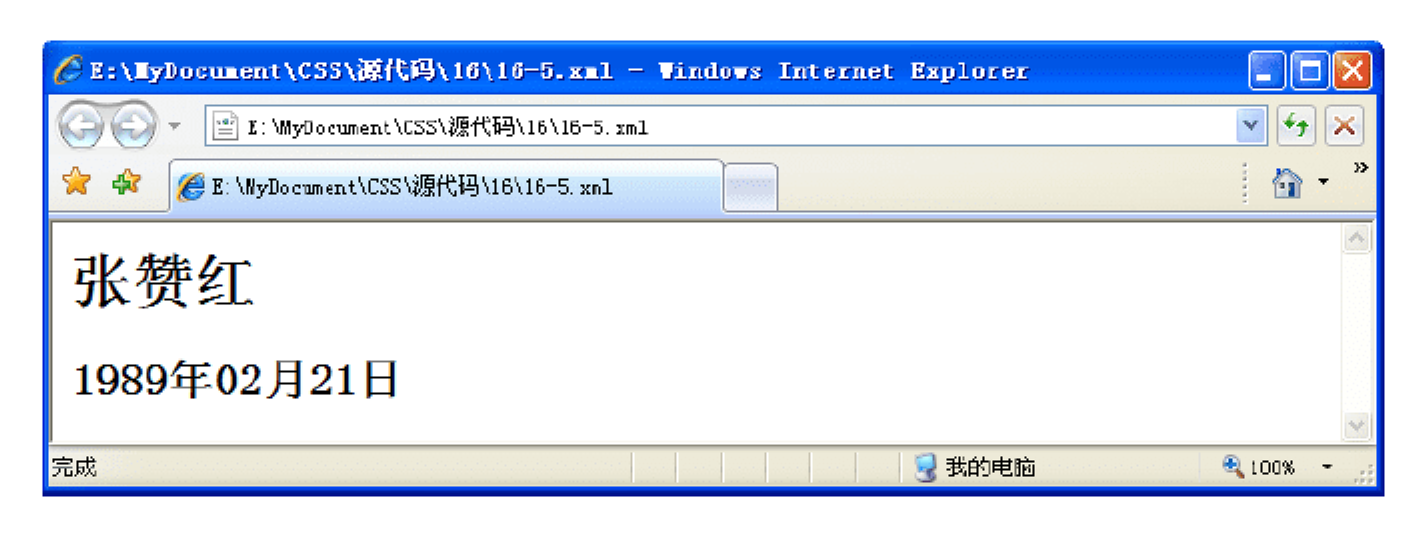


图 16-5　使用 XSLT 转换显示 XML 中的数据

16.5.2　实例描述

本人比较喜欢宠物狗，今天闲来无事，就去网上逛了逛宠物网站，宠物网站就是漂亮，各种各样的宠物在页面上展现出来真是美到极点啦！做程序员的看网站不是那么简单地饱饱眼福就行了，还需要研究别人是怎么做的，于是我将页面保存下来，研究了半天才发现，人家是使用 XSLT 将 XML 文档转换为 HTML 页面来显示 XML 数据的。下面我们也来试着做一下吧！

16.5.3　实例应用

【例 16-5】使用 XSLT 制作宠物展示页面

(1) 创建 16-6.xml 文件，提供宠物信息数据，内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<动物一览表>
    <比熊>
            <名称>比熊</名称>
            <图片>images/1.jpg</图片>
    </比熊>
    <贵宾犬>
            <名称>贵宾犬</名称>
            <图片>images/2.jpg</图片>
    </贵宾犬>
    <山东狮子猫>
            <名称>山东狮子猫</名称>
            <图片>images/3.jpg</图片>
    </山东狮子猫>
    <雪鞋猫>
            <名称>雪鞋猫</名称>
            <图片>images/4.jpg</图片>
    </雪鞋猫>
</动物一览表>
```

(2) 创建 XSLT 文件，命名为 16-6.xsl。

(3) 在 16-6.xsl 文件中使用 XSLT 格式在页面中生成表格，在表格中显示动物信息，文件内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<!-- 设置动物图片-->
<xsl:attribute-set name="image">
    <xsl:attribute name="width">150</xsl:attribute>
    <xsl:attribute name="height">150</xsl:attribute>
</xsl:attribute-set>
<!-- 指定输出到页面的编码方式为 GBK-->
<xsl:output method="html" encoding="GBK"/>
<!-- 根标记模式-->
<xsl:template match="/">
    <!-- 指定背景色-->
    <body background="images/5c7db781ad9030aff703a65a.jpg"></body>
    <!--使用 xsl:element 代替<html>标签 -->
    <xsl:element name="html">
        <xsl:element name="table">
            <!--指定 table 元素的 border 属性 -->
            <xsl:attribute name="border">1</xsl:attribute>
            <!--循环遍历"动物一览表"标记下的元素 -->
            <xsl:for-each select="动物一览表">
                <xsl:apply-templates/>
           </xsl:for-each>
       </xsl:element>
   </xsl:element>
```

```
</xsl:template>
<xsl:template match="动物一览表/*">
            <xsl:element name="td">
                <!-- 指定 td 元素的 width 属性-->
                <xsl:attribute name="width">90</xsl:attribute>
                <xsl:value-of select="名称"/>
              </xsl:element>
              <xsl:element name="td">
                <!-- 引用上面定义好的 image 标记属性-->
                <xsl:element name="image" use-attribute-sets="image">
                    <xsl:attribute name="src">
                        <xsl:value-of select="图片"/>
                     </xsl:attribute>
                 </xsl:element>
            </xsl:element>
</xsl:template>
</xsl:stylesheet>
```

(4) 在 16-6.xml 文档中添加引用 XSLT(16-6.xsl)文件，代码如下：

```
<?xml-stylesheet href="16-6.xsl" type="text/xsl"?>
```

16.5.4　运行结果

在浏览器中打开 16-6.xml 文件，在页面中以表格的形式展示宠物，如图 16-6 所示。

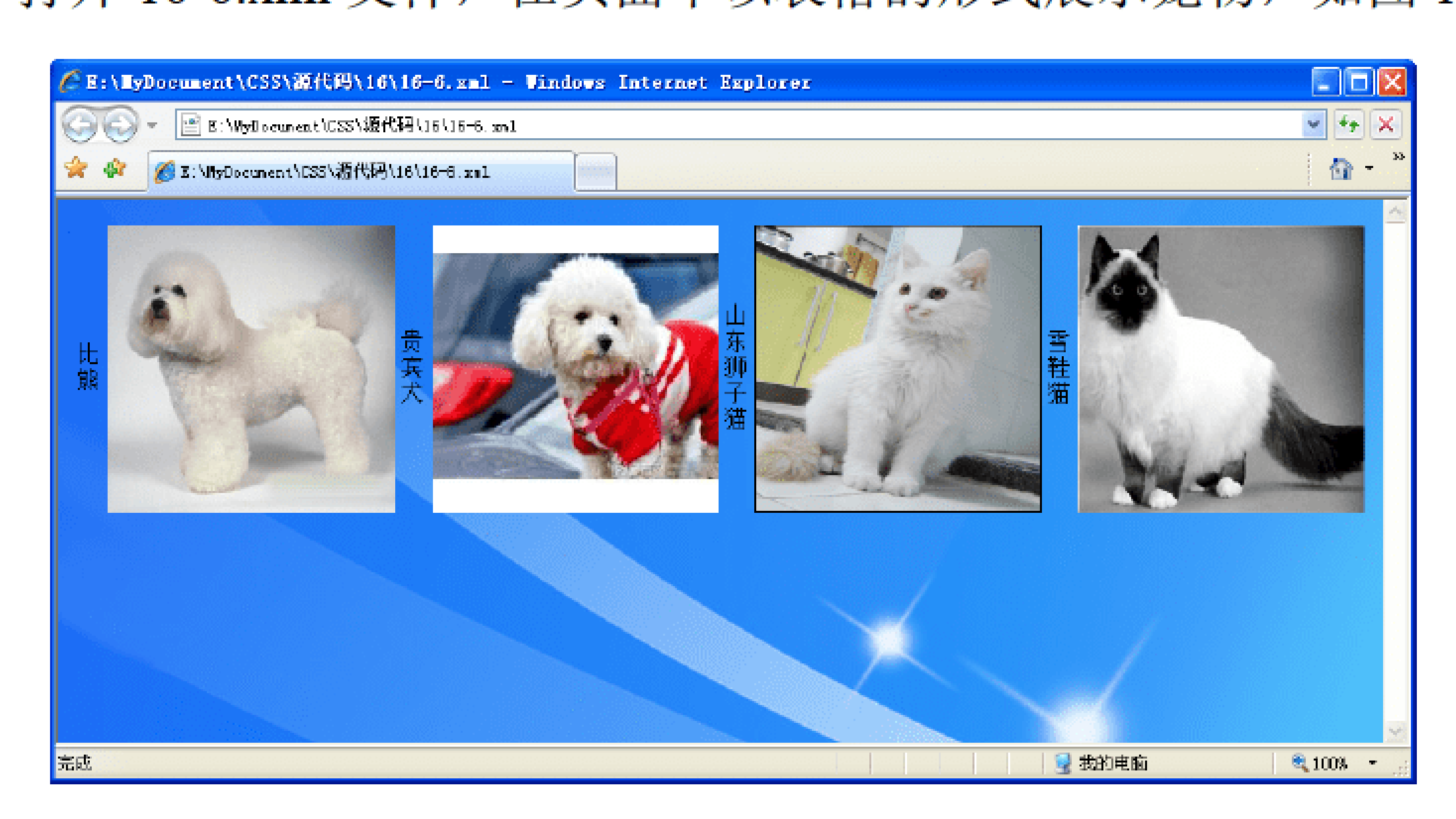


图 16-6　宠物展示

16.5.5　实例分析



源码解析：

在这个实例中，我们使用了 XSL 标记“xsl:element”来代替模板中的 html 元素，使用“xsl:attribute”来指定元素的属性；使用<xsl:for-each select="动物一览表">”来指定要循环的

是“动物一览表”下的标记元素；使用“<xsl:template match="动物一览表/*">”指定标记匹配模式是“动物一览表”标记下的所有子标记，从而将“比熊”、“贵宾犬”、“山东狮子猫”、“雪鞋猫”标记中的数据输出到表格中显示在页面上。

16.6 常见问题解答

16.6.1 设置的 CSS 样式无效的问题



设置的 CSS 样式无效的问题。
网络课堂：http://bbs.itzcn.com/thread-12278-1-1.html

我编辑了一个 CSS 样式文件 text.css，设置了一些样式规则，出现在页面上的“NAME1”应该是 blue，“NAME2”应该是 red，但是，出现的字体颜色总是 blue，我编辑的 CSS 样式文件如下所示：

```
name {font-size:24px;color:blue}
name #a1{font-size:24px;color:red}
```

编辑的 XML 文件内容如下所示：

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/css" href="css/text.css"?>
<student>
  <name>NAME1</name>
  <name ID="a1">NAME2</name>
</student>
```

我仔细检查了一遍，没什么问题，这是怎么回事啊？

【解决办法】仔细查看 CSS 样式文件中的“name #a1”，“name”与“#a1”之间有一个空格，这个空格是不允许存在的，如果加空格的话，XML 解析器无法解析你所编辑的“name #a1”是设置 XML 文档中的 ID 为“a1”的 name 标记，而是当做一个普通样式来对待。因此修改 CSS 样式文件中的“name #a1”为“name#a1”即可。

16.6.2 为什么还要发展 XML



为什么还要发展 XML？
网络课堂：http://bbs.itzcn.com/thread-12280-1-1.html

HTML 已经很流行了，为什么还要使用 XML 来显示数据呢？

【解决办法】简单地讲，HTML 不能完成我们希望 XML 完成的任务。原因很简单，XML 所要完成的任务，必须由元置标语言来完成，而 HTML 只是一个实例置标语言。在 XML 发布之前，国际互联网的发展受到 HTML 如下几个问题的束缚。

(1) HTML 无法描述数据内容，而这一点恰恰是数据检索、电子商务所必需的。

(2) HTML 对数据表现的描述能力是十分不够的，如 HTML 还不能描述矢量图形、科学符号等对象，目前只能通过图像来表现这些对象。

(3) HTML 实例置标语言的地位，完全不能适应对新标记需求的发展需要。

XML 的出现，使上述问题都得到了很好的解决。试想在互联网世界，如果大家都讲方言，互相交换信息时都要进行翻译，那将是一个多么难以沟通和交流的世界。在互联网世界中，XML 之所以重要，恰恰是由于它扮演了“国际语言”的角色。此外，XML 还提供了用于定义各行各业的“专业术语”的工具。

16.6.3　为什么要使用 XML 而不是 HTML



为什么要使用 XML 而不是 HTML?

网络课堂：http://bbs.itzcn.com/thread-12282-1-1.html

有些情况，为什么要用 XML 文档而不用 HTML 文档啊?

【解决办法】比较重要的原因有以下六条。

(1) 供应商能使用 XML 设计自己的文档类型，而 HTML 只能设计 HTML 类型的文档。

(2) XML 的超文本链接能力比 HTML 强得多，XML 提供的信息内容比 HTML 更丰富，也更易于使用。

(3) XML 能提供更多更好的机制方便浏览器的信息表现和优化性能。

(4) XML 舍弃了 SGML 的复杂性，因此编写处理 XML 的应用程序会很容易。

(5) 易于存储，可重复使用。

(6) XML 文件在 SGML 环境中也可使用，不一定要局限于在 Web 中使用。

16.6.4　XML 如何与数据库连接



XML 如何与数据库连接?

网络课堂：http://bbs.itzcn.com/thread-12284-1-1.html

在 Web 开发中，与数据库连接是必然的，那么使用 XML 如何与数据库连接啊?

【解决办法】XML 是一种文件格式，它没有规定与数据库的连接方法，你需要用传统的方法连接数据库，进行数据库查询，然后将查询结果转换为 XML 格式。现在有一些工具提供的 XML 与数据库的连接过程大都遵循这样的步骤。下面是一个利用 ASP 直接生成 XML 文件的例子：

```
<%@ language="VBScript"%>
<?xml version="1.0" encoding="gb2312"?>
<?xml:stylesheet type="text/xsl" href="..image oster.xsl"?>
<roster>
    <%
        set cConn = Server.CreateObject("ADODB.Connection")
```

```
        call cConn.Open("DSN","USER", "PWD")
        set rs = cConn.Execute("SELECT DISTINCT * FROM roster")
        Do While Not rs.EOF
    %>
    <Record>
        <Name><%=trim(rs("name"))%></Name>
        <NativePlace><%=trim(rs("NativePlace"))%></NativePlace>
        <Age><%=trim(rs("Age"))%></Age>
        <Telephone><%=trim(rs("Telephone"))%></Telephone>
    </Record>
    <% rs.MoveNext
        Loop
        rs.Close
        set rs=nothing
        set cConn=nothing
    %>
</roster>
```

16.7 习　　题

1. 填空题

(1) XML 文档结构可以分为三个部分：__________、主体和尾声。

(2) XML 文档中的主体就是 XML 文档描述数据的地方，通常由标记、元素和__________等构件组成。

(3) 下面 XML 文档内容中，<姓名>是开始标记、</姓名>是结束标记，而<姓名>李赞红</姓名>就构成了__________。

```
<?xml version="1.0" encoding="UTF-8"?>
    <学生>
        <姓名>李赞红</姓名>
        <性别>男</性别>
        <专业>计算机</专业>
    </学生>
```

(4) 在 XML 文档中嵌入 CSS 样式需要在__________标签的开始与结束之间编辑 CSS 样式。

(5) 一个样式表的格式如下：

```
文本代表
{
    样式规则
}
```

对于 XML 文件，样式表中的“文本代表”可以是标记的名称，也可以是__________。

(6) XML 文件需要使用__________操作指令将 XML 文件和 XSLT 样式表关联。

2. 选择题

(1) 将 CSS 与 XML 文档相联系的语法是＿＿＿＿＿＿。

A.

```
<?xml-stylesheet href="xxx.css" type="text/css"?>
```

B.

```
<?stylesheet href="xxx.css" type="text/css"?>
```

C.

```
<xml-stylesheet href="xxx.css" type="text/css">
```

D.

```
<?xml-stylesheet "xxx.css" type="text/css"?>
```

(2) XML 文档中的一个非空元素是由＿＿＿＿＿＿构成的。

A. 开始标记、结束标记　　B. 开始标记、结束标记及两标记之间的数据

C. 标记　　D. 标记之间的数据

(3) 在 XML 文档中使用 CSS 样式正确的方式为＿＿＿＿＿＿。

A. 引用式和嵌入式　　B. 内部和外部

C. 交叉引用　　D. 包含式

(4) 使用 XSLT 定义 XML 文档显示方式的基本思想是＿＿＿＿＿＿。

A. 通过定义转换模板，将 XML 源文档转换为带样式信息的可浏览文档。

B. 定义不同以往的显示风格。

C. 控制 XML 文档显示数据的输出。

D. 通过定义显示模板，显示指定的 XML 数据。

(5) XML 文档中有下面一段代码:

```
<teacher>
    <name id="tea1">孔子</name>
    <name id="tea2">老子</name>
    <name id="tea3">老舍</name>
</teacher>
```

假设需要把 ID 为 “tea1” 的 name 标记字体显示为红色，ID 为 “tea2” 的 name 标记字体显示为绿色，ID 为 “tea3” 的 name 标记字体显示为蓝色，则 CSS 样式怎么定义？

＿＿＿＿＿＿

A.

```
name #tea1{color:red;}
name #tea2{color:green;}
name #tea3{color:blue;}
```

B.

```
tea1{color:red;}
```

```
tea2{color:green;}
tea3{color:blue;}
```

C.

```
name {color:red;
    color:green;
    color:blue;
}
```

D.

```
name#tea1{color:red;}
name#tea2{color:green;}
name#tea3{color:blue;}
```

3. 上机练习

上机练习：设计笔记本推荐网页

要求：

(1) 在页面中显示各个品牌(戴尔、惠普、三星)电脑的详细信息，以表格的形式展现在页面中。各个品牌电脑的详细信息包括电脑的品牌、价格、类型、CPU、内存大小信息。

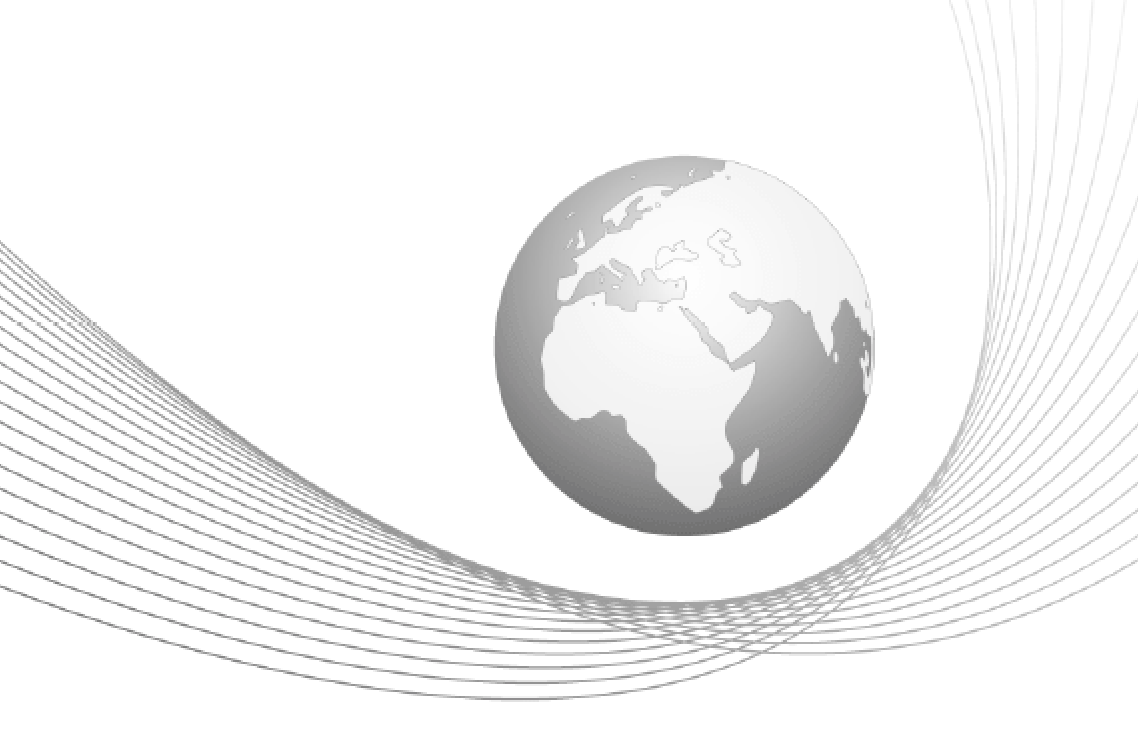
(2) 设置标题的字体大小为 40 像素、加粗并居中显示；设置表格背景色为粉色，表格居左显示，单元格中的字体为 15 像素；设置电脑品牌字体为红色、价格字体为紫色、类型字体为蓝色、CPU 字体为绿色和内存字体为深蓝色。

(3) 在页面的底部展示三张电脑图片。

最后的运行效果如图 16-7 所示。



图 16-7 笔记本推荐网页



第 17 章　尽善尽美——合理使用 JavaScript

内容摘要：

对于传统的 HTML 语言而言，很难开发出具有动态和交互性的页面，而使用 CSS 除了可以轻松设置网页元素的显示位置和格式外，甚至还能产生滤镜，图像淡化，网页的淡入淡出的渐变效果。同样 CSS 可以实现的效果使用 JavaScript 也可实现，例如：设置表格边框的颜色、使用滤镜实现图片的翻转等。

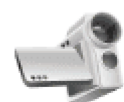
本章将 CSS 与 JavaScript 相结合实现各种页面的绚丽效果，例如：使表格的边框变换颜色、随着鼠标浮动的文字、会闪的图片等。

学习目标：

- 掌握如何结合 JavaScript 制作更炫的表格边框
- 实现文字可以随鼠标浮动
- 了解如何使图片会不断闪烁
- 理解如何使图片翻转
- 理解如何实现文字可以在页面中舞动
- 掌握如何实现导航栏可收缩
- 理解如何实现表格会逐渐变色

17.1 会变色的表格边框

在前面章节中已经详细介绍过如何为页面元素添加边框，但前面只是使用 CSS 简单设置了一下元素边框的样式、颜色。本节通过应用 JavaScript 编程语言，可以使表格元素的边框不断变色，从而使整个表格看起来更加绚丽。

**视频教学**：光盘/视频/17/ 17.1 会变色的表格边框.avi 长度：5 分钟

17.1.1 实例描述

在上网浏览网页时，经常会看到这种效果：在一些需要来访者注意的地方或是该网页的亮点部分会被用高亮或变色的边框框起来。本节实例将实现这种效果，为了使表格引起浏览者的特别注意，将表格的边框设置为会变色的样式，效果非常酷哦，快来看看吧。

17.1.2 实例应用

【例 17-1】会变色的表格边框

(1) 新建一个 border-color.html 文件，在该文件中首先定义一个表格，代码如下所示：

```
<table border="0" width="280" id="myexample" style="border:5px solid green"
align="center">
    <tr>
        <td>
            <p align="center">
                Java Script 是一种基于对象和事件驱动并具有相对安全性的客户端脚本语言。
            </p>
        </td>
    </tr>
</table>
```

(2) 使用 JavaScript 脚本设置表格边框的颜色变化，代码如下所示：

```
<script language="Java Script">
    function flashit(){
        if (!document.all)  <!--判断是否是 IE 浏览器-->
            return
        if (myexample.style.borderColor=="green")  <!-- 判断当前表格边框的颜色
是否为 green -->
            myexample.style.borderColor="red"  <!-- 当前表格边框颜色为 green，将
表格边框设为 red -->
        else                <!-- 否则的话，设为 green -->
            myexample.style.borderColor="green"
    }
```

```
    setInterval("flashit()", 500)       <!-- 每 500 秒调用一次 flashit()函数 -->
</script>
```

17.1.3　运行结果

在 IE 浏览器中打开 border-color.html 文件，执行效果如图 17-1 所示。

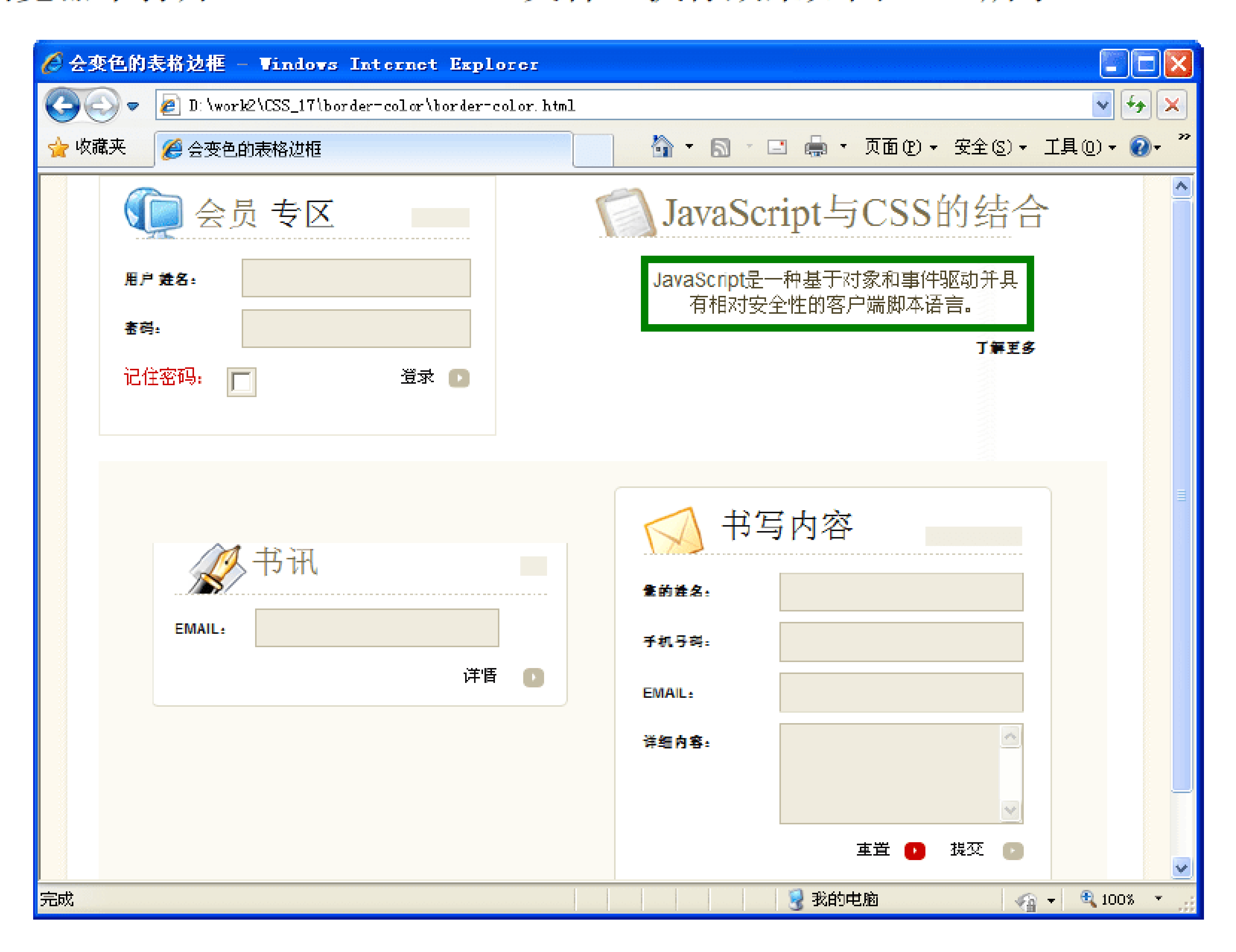


图 17-1　会变色的表格边框

17.1.4　实例分析



源码解析：

本实例首先在页面中定义了一个表格，并为表格设置了边框，初始边框颜色为 green，然后通过 JavaScript 脚本 myexample.style.borderColor 设置边框的颜色为红色，并通过判断当前边框的颜色来切换设置边框的颜色，最后调用 setInterval("flashit()", 500)函数，每 500 秒调用一次 flashit()函数。这样会变色的表格边框就形成了。

17.2　随着鼠标浮动的文字

在网上看到很多鼠标特效，后来才知道这些特效可以采用 JavaScript 来实现，就研究了一下。本节就演示一下我的收获，共同学习实现文字随鼠标浮动的效果。

 视频教学：光盘/视频/17/17.2　随着鼠标浮动的文字.avi　　长度：4 分钟

17.2.1　实例描述

前天在浏览一个朋友的博客时发现，鼠标后面会跟着浮动一些文字，看起来很炫，现在自己也实现了这种炫的鼠标特效，就拿出来与大家分享一下。通过本实例的学习，要求读者掌握这种鼠标特效的实现原理。

17.2.2　实例应用

【例 17-2】随着鼠标浮动的文字

新建一个 text-font.html 文件，在该文件中定义一段文本，设置一下文本的字体、颜色，并使该文本可以随着鼠标浮动，代码如下所示：

```
<script language="Java Script">
    if (document.all)
    {
        yourLogo = "圣诞节快乐！！！";          <!--待浮动的文字-->
        LogoFont = "Arial";                    <!--文字的字体-->
        LogoColor = "red";                     <!--文字的颜色-->
        yourLogo = yourLogo.split('');         <!--将浮动的字符串分成单个的字符-->
        L = yourLogo.length;                   <!--获得字符串的长度-->
        TrigSplit = 360 / L;                   <!--设置每次旋转的角度-->
        Sz = new Array()                       <!--声明一个数组-->
        LogoWidth = 100;                       <!--定义宽度-->
        LogoHeight = -30;                      <!--定义高度-->
        ypos = 0;
        xpos = 0;
        step = 0.03;                                 <!--定义步长-->
        currStep = 0;
        document.write('<div id="outer"
        style="position:absolute;top:0px;left:0px"><div
        style="position:relative">');
        for (i = 0; i < L; i++)
        {
            document.write('<div id="ie"
            style="position:absolute;top:0px;left:0px;'+'width:10px;

    height:10px;font-family:'+LogoFont+';font-size:12px;'+'color:'+LogoColor+';
            text-align:center">'+yourLogo[i]+'</div>'); <!--依次显示每一个字符-->
        }
        document.write('</div></div>');
        function Mouse()
        {
            ypos = event.y;                    <!--获得当前鼠标的纵坐标-->
```

```
            xpos = event.x - 5;             <!--获得当前鼠标的横坐标-->
        }
        document.onmousemove=Mouse;         <!--鼠标移动后调用 mouse 函数-->
        function animateLogo()
        {
            <!--获得当前窗口的坐标-->
            outer.style.pixelTop = document.body.scrollTop;
            for (i = 0; i < L; i++)
            {
                ie[i].style.top = ypos + LogoHeight * Math.sin(currStep + i *
                 TrigSplit * Math.PI / 180);    <!--获得第 i 个字符的上边界-->
                ie[i].style.left = xpos + LogoWidth * Math.cos(currStep + i *
                TrigSplit * Math.PI / 180);     <!--获得第 i 个字符的左边界-->
                Sz[i] = ie[i].style.pixelTop - ypos;    <!--设置字体-->
                if (Sz[i] < 5) Sz[i] = 5;               <!--设置字体-->
                ie[i].style.fontSize = Sz[i] / 1.7;     <!--当前字体的变化-->
            }
            currStep -= step;
            setTimeout('animateLogo()', 20);    <!--动画的实现-->
        }
        window.onload = animateLogo;            <!--直接调用 animateLogo 函数-->
    }
</script>
```

17.2.3　运行结果

在 IE 浏览器中运行 text-font.html 文件，执行效果如图 17-2 所示。



图 17-2　随着鼠标浮动的文字

17.2.4 实例分析

**源码解析：**

首先定义浮动文字的内容、字体、字体颜色，并将浮动的字符串分成单个的字符，获得字符串的长度。设置字符每次旋转的角度、宽度和高度。

然后使用 JavaScript 脚本将定义好的文本放到 div 元素中，并显示在页面中，还定义了一个获取鼠标坐标的函数 Mouse()，当鼠标移动时触发 document.onmousemove 指定的 Mouse()函数，通过得到的鼠标坐标来设置字符显示的位置。并且通过 setTimeout('animateLogo()', 20)函数每 20 秒调用一次 animateLogo()函数。

最后指定 window.onload=animateLogo()，打开窗口直接执行 animateLogo()函数。这样就实现了文字随鼠标浮动的效果。

17.3 会闪的图片

有很多使用 CSS 来实现的样式，实际上使用 JavaScript 也可以实现，本节将使用 JavaScript 脚本语言来设置图片的显示与隐藏属性，使图片一隐一现闪烁起来吧。

**视频教学：**光盘/视频/17/17.3　会闪的图片.avi　　　　长度：3 分钟

17.3.1 实例描述

刚在网上商城逛了一圈，看到有些商品的图片特效做得相当不错，就想做一些图片特效放到自己的博客中。最后，就实现了本文所介绍的图片不停闪烁的特效。放到博客上之后，为我的博客吸引了不少浏览者，大家也来试试吧。

17.3.2 实例应用

【例 17-3】会闪的图片

(1) 新建一个 picture.html 文件，在该文件中定义一个 div 元素，并包含一个精美图片，代码如下所示：

```
<body onLoad="pictureOnload()" topmargin="0">
    <div id="picture" >
        <img src="pic_files/1265627301_14441ce9.jpg" >
    </div>
</body>
```

(2) 使用 JavaScript 脚本语言设置图片的 visibility 属性，使图片一隐一现，这样图片就闪

起来了，代码如下所示：

```
<script language="Java Script">
    var ms = 800;           //定义一个时间
    var counter = 0;            //统计变量
    function pictureOnload() {
        setTimeout("style fun()", ms);
    }
    function style_fun() {
        picture.style.visibility =
        (picture.style.visibility == "hidden") ? "visible" : "hidden";
                                                    //判断并设置图片的 visibility 属性
        counter +=1;
        setTimeout("style_fun()", ms);     //每 800 秒调用一次 style_fun()函数
    }
</script>
```

17.3.3　运行结果

在 IE 浏览器中运行 picture.html 文件，执行效果如图 17-3 所示。



图 17-3　会闪的图片

17.3.4　实例分析



源码解析：

本实例首先在页面中定义一个包含图片的 div 元素。然后使用 JavaScript 脚本判断图片的 visibility 属性是否为 hidden。如果为 hidden 表示当前图片为隐藏状态，将图片的 visibility 属性

设为 visible，使图片显示出来，否则的话，将图片设为 hidden 隐藏状态。最后用 setTimeout("style_fun()", ms)每 800 秒调用一次 style_fun()函数，这样图片就出现了闪烁的效果。

17.4 图 片 翻 转

前面章节已经介绍了如何使用滤镜制作一些特殊效果，实际上使用 JavaScript 也完全可以实现滤镜的这些效果，本节将使用 JavaScript 脚本语言来实现图片翻转的功能。

**视频教学：**光盘/视频/17/17.4 图片翻转.avi　　长度：4 分钟

17.4.1 实例描述

好久没有进过论坛了，利用午休时间去看看吧。到论坛中一看又有不少求助的帖子，突然看到有人问如何使用 JavaScript 实现图片翻转的效果。正好本人研究过这个东西，就帮他实现了一个可以水平和垂直翻转图片的实例，今天也借此机会和大家一起分享一下吧。

17.4.2 实例应用

【例 17-4】图片翻转

(1) 新建一个 reverse.html 文件，在该文件中定义一个图片元素和一个 form 表单元素，通过单击表单按钮，将图片进行水平翻转和垂直翻转，代码如下所示：

```
<body>
    <div ><img src="4.gif"  id="obj"></div>
    <form id="flip_form" name="flip_form">
        使用滤镜实现图片翻转效果:
        <input type="button" value="水平翻转" onClick="setFliph()">
        <input type="button" value="垂直翻转" onClick="setFlipv()">
    </form>
</body>
```

(2) 使用 JavaScript 脚本实现滤镜翻转图片的效果，代码如下所示：

```
<script language="Java Script">
    function setFliph(){
        obj.style.filter=obj.style.filter=="fliph"?"":"fliph";
                                //设置 obj 的 filter 样式为 fliph，水平翻转图片
    }
    function setFlipv(){
        obj.style.filter=obj.style.filter=="flipv"?"":"flipv";
                                //设置 obj 的 filter 样式为 flipv，垂直翻转图片
    }
</script>
```

17.4.3　运行结果

在 IE 浏览器中运行 reverse.html 文件，可以看到图片下方有两个按钮，分别是“水平翻转”和“垂直翻转”，当单击“水平翻转”按钮时，图片水平翻转，执行效果如图 17-4 所示。

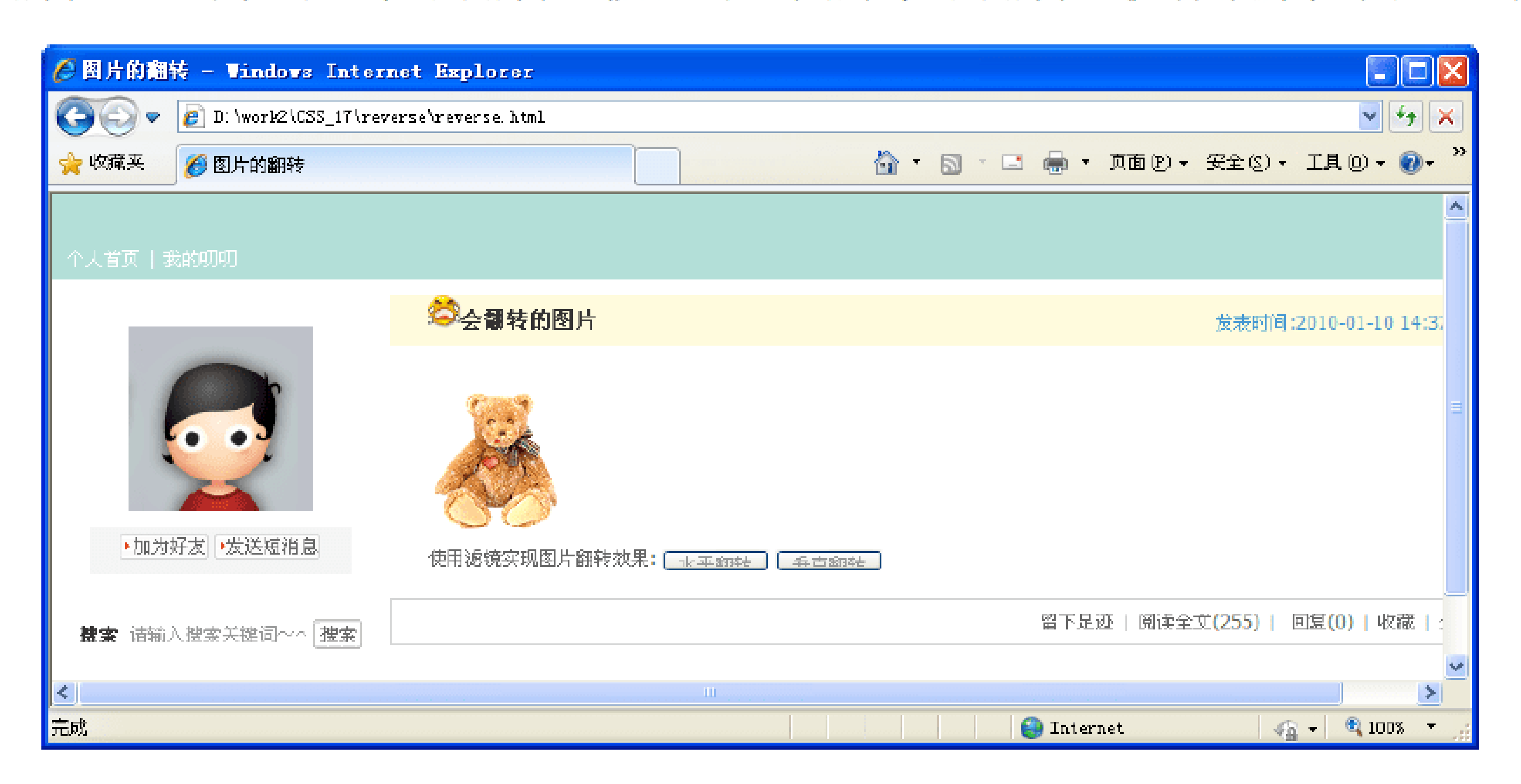


图 17-4　水平翻转

单击“垂直翻转”按钮时，图片进行垂直翻转，执行效果如图 17-5 所示。

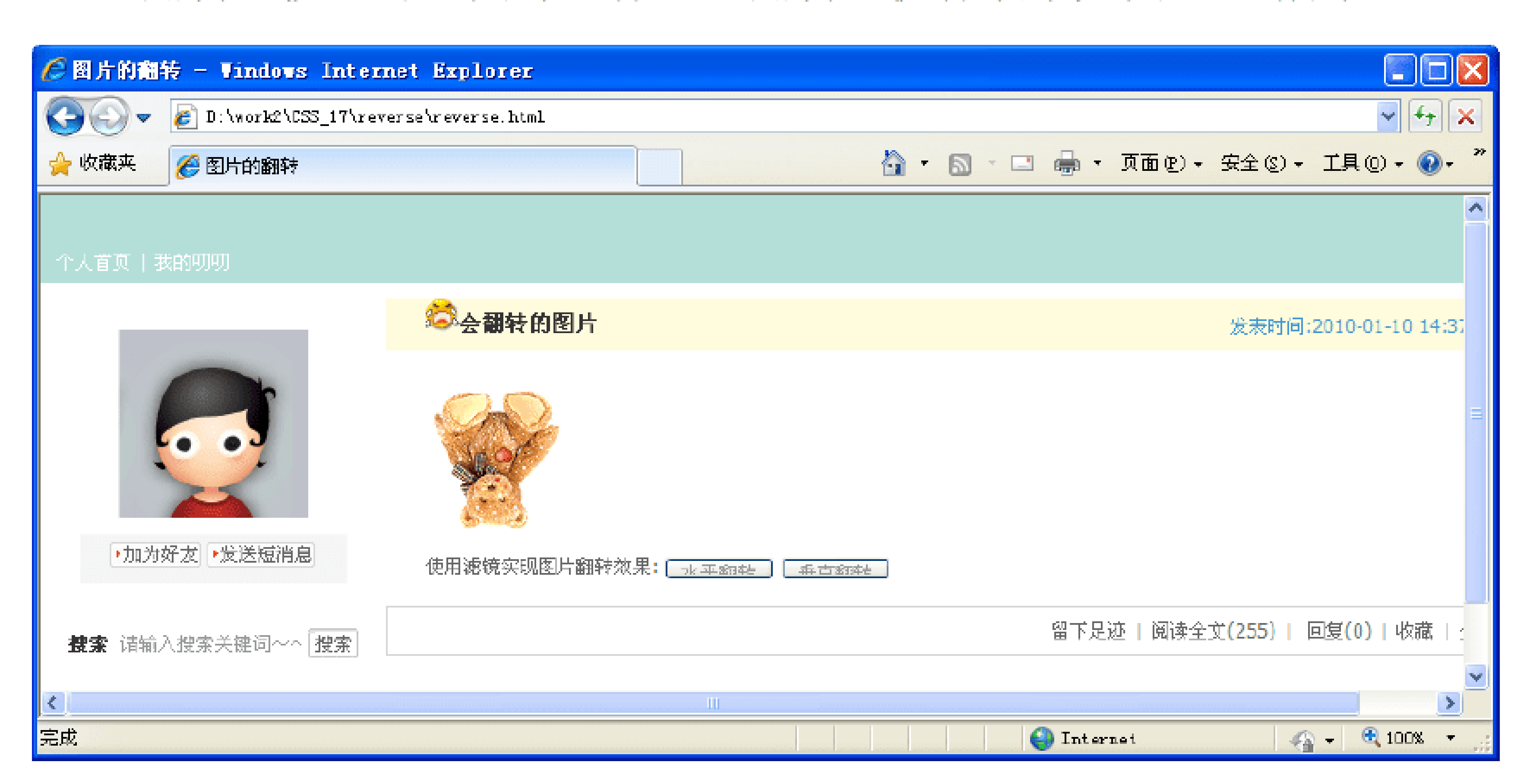


图 17-5　垂直翻转

17.4.4　实例分析



源码解析：

本实例首先在 reverse.html 页面中定义了一个包含 img 元素的 div，然后定义了一个 form 表单元素，该表单元素包含两个按钮，用来控制图片进行翻转，分别是“水平翻转”和“垂直翻转”。

最后定义了两个函数，在 setFliph()函数中设置 obj(即指图片 id)的 filter 样式为 fliph 使图片水平翻转，setFilpv()函数中设置 obj 的 filter 样式为 flipv 使图片垂直翻转。这两个函数分别为“水平翻转”和“垂直翻转”按钮的单击触发事件函数。

17.5　舞动的文字

一些绚丽的效果仅仅使用 CSS 常常是无法实现的，比如本节要实现文字在页面中舞动的效果。但是，如果结合 JavaScript 脚本进行实现，那么就简单多了。

 **视频教学：**光盘/视频/17/17.5　舞动的文字.avi　　长度：4 分钟

17.5.1　实例描述

经常上网的人应该看到过很多网页文字特效，例如：跑马灯、文字垂直滚动、从页面顶部落下等。本节实例将演示文字在页面中舞动的效果。

17.5.2　实例应用

【例 17-5】舞动的文字

(1) 新建一个 HTML 文件，命名为 text_dance.html，定义在该文件中显示的文字内容、字体、字号、颜色等，设置文字在浏览器中显示的各个边界，还需初始化每个字符的 x、y 坐标，以及文字在页面显示的格式。代码如下所示：

```
<script>
    var your_message="新年快乐！！！"       <!--显示文字的内容-->
    var textfont="Verdana"                  <!--显示字体-->
    var textsize=20                         <!--显示字号-->
    var textcolor="red"                     <!--颜色设定-->
    var textboldness=15                     <!--字体粗细-->
    var tempo=200                           <!--延时长短-->
    var stepx=20                            <!--移动的步长，分水平和垂直方向的-->
    var stepy=15
    var msg=new Array()
    var msg=your_message                    <!--将要显示的内容赋给 msg-->
    var message=msg.split("")               <!--将 msg 在空格处拆分-->
    var numberofletters=msg.length-1        <!--字符的个数-->
    var x,y
    var marginbottom                        <!--浏览器的底部-->
    var marginleft=0                        <!--浏览器的左边界-->
    var margintop=0                         <!--浏览器的顶部-->
    var marginright                         <!--浏览器的右边界-->
```

```
    textboldness=textboldness*100        <!--字体粗细设定-->
    var timer                            <!--计时-->
    var xpos=new Array()                 <!--横、纵坐标-->
    var ypos=new Array()
    var spancontent=new Array()          <!--要显示的内容，包括格式的设定-->

    for (i=0; i<=numberofletters;i++) {  <!--初始化，每个字符的 x、y 坐标均为 0-->
        xpos[i]=0
        ypos[i]=0
    }

    for (i=0;i<=numberofletters;i++) {  <!--按照指定的格式显示字符文字-->
        spancontent[i]="<span
style='position:relative;font-family:"+textfont+";font-size:"+
        textsize+"pt;color:"+textcolor+";font-weight:"+textboldness+"'>"+
        message[i]+"</span>"
    }
</script>
```

(2) 定义一个获取随机数的函数 randommaker()，用于更新文字移动的步长，代码如下所示：

```
<script>
    function randommaker(range) {                <!--产生随机数-->
        rand=Math.floor(range*Math.random())
        return rand
    }
</script>
```

(3) 定义 checkposition()函数，用于控制文字移动不超出浏览器的边界，代码如下所示：

```
<script>
    function checkposition() {
        if (document.all) {              <!--如果是 IE 浏览器-->
            if (document.all.span0.style.posLeft>marginright) {
                                         <!--如果位置超出右边界-->
                stepx=(stepx+randommaker(2))*-1        <!--更新横坐标步长值-->
                document.all.span0.style.posLeft-=1    <!--posLeft 减 1-->
            }
            if (document.all.span0.style.posLeft<marginleft) {
                                         <!--如果位置超出左边界-->
                stepx=(stepx+randommaker(2))*-1       <!--更新横坐标步长值-->
                document.all.span0.style.posLeft+=1   <!--posLeft 加 1-->
            }
            if (document.all.span0.style.posTop>marginbottom) {
                                         <!--如果位置超过底部-->
                stepy=(stepy+randommaker(2))*-1       <!--更新纵坐标步长值-->
                document.all.span0.style.posTop-=1    <!--posTop 减 1-->
            }
```

```
            if (document.all.span0.style.posTop<margintop) {  <!--如果位置超出顶部-->
                stepy=(stepy+randommaker(2))*-1      <!--更新纵坐标步长值-->
                document.all.span0.style.posTop+=1   <!--posTop 加 1-->
            }
        }

    }
</script>
```

(4) 定义使文字移动的函数 moveball()，在该函数中调用 checkposition()函数检查文字位置是否超出边界，并给予必要的调整；然后调用 makesnake()函数使文字移动飞舞起来，代码如下所示：

```
<script>
    function moveball() {                    <!--使文字移动的函数-->
        if (document.all) {                  <!--如果是 IE 浏览器-->
            checkposition()
                                <!--调用 checkposition 函数,检查位置并给予必要的调整-->
            makesnake()         <!--调用 makesnake 函数-->
            document.all.span0.style.posTop+=stepy
                                <!--改变垂直方向上的位置值-->
            timer=setTimeout("moveball()",tempo)       <!--延时 40ms-->
        }

    }

    function makesnake() {
        for (i=numberofletters; i>=1; i--) {
                                <!--循环，使得第 i 个字符占据第 i—1 个字符的位置-->
            xpos[i]=xpos[i-1]
            ypos[i]=ypos[i-1]
        }
        if (document.all) {                  <!--如果是 IE 浏览器-->
            xpos[0]=document.all.span0.style.posLeft+stepx
                                <!--确定第 1 个字符的位置-->
            ypos[0]=document.all.span0.style.posTop+stepy
            for (i=0;i<=numberofletters;i++) {          <!--对于每个字符-->
                var thisspan=eval("document.all.span"+(i)+".style")
                thisspan.posLeft=xpos[i]
                                <!--将更新后的 x、y 坐标赋给 thisspan-->
                thisspan.posTop=ypos[i]
            }
        }
    }
</script>
```

(5) 定义 setValues()函数，使页面执行时，文字延时 2 秒钟出现。setValues2()函数设定文

字在浏览器中显示的底边和右边，并将文字初始显示位置设为页面的左上角，代码如下所示：

```
<script>
    function setValues() {          <!--设定延时为 2s-->
        var firsttimer= setTimeout("setValues2()",2000)
    }

    function setValues2() {
        if (document.all) {         <!--如果是 IE 浏览器-->
            marginbottom = document.body.clientHeight-5;
                                    <!--设定显示的底边和右边-->
            marginright = document.body.clientWidth-5
            for (i=0;i<=numberofletters;i++) {
                var thisspan = eval("document.all.span"+i)
                thisspan.innerHTML=spancontent[i]
                                    <!--将 spancontent 的值返回给 thisspan-->
                var thisspan = eval("document.all.span"+(i)+".style")
                thisspan.posLeft=0  <!--将位置确定为左上角-->
                thisspan.postop=0
            }
            moveball()              <!--文字开始移动-->
        }
    }
</script>
```

(6) 通过 for 循环在页面中显示定义的文字内容，代码如下所示：

```
<script>
    for (i=0;i<=numberofletters;i++) {      <!--显示上面定义的字符串-->
        document.write("<span id='span"+i+"'
        style='position:absolute'></span>")
        document.close()                    <!--关闭显示-->
    }
</script>
```

(7) 将 setValues()函数设置加载到 body 元素上，使文字在整个 body 中舞动显示，代码如下所示：

```
<body  onload=setValues()>
</body>
```

17.5.3　运行结果

在 IE 浏览器中运行 text_dance.html 文件，可以看到“新年快乐！！！”文字在页面中舞动，执行效果如图 17-6 所示。



图 17-6　舞动的文字

17.5.4　实例分析

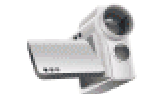


源码解析：

本实例首先定义了在页面中显示的文字内容、字体、颜色等变量。定义获取随机数的函数randommaker()用于更新文字移动的步长。定义文字移动的函数 moveball()，在该函数中调用checkposition()函数检查文字位置是否超出边界，通过 makesnake()函数使文字舞动起来。最后调用 setValues()函数使页面执行时，文字延迟 2 秒钟出现，并且设定文字初始显示位置在页面的左上角。这样“新年快乐！！！”这几个文字就在页面中舞动起来了。

17.6　会收缩的导航栏

用 JavaScript 实现一个网页特效实在是太方便了，本节我们就使用 JavaScript 来实现一个会收缩的导航栏，为我们的页面增光添彩，而且使用也非常方便。

**视频教学：**光盘/视频/17/17.6　会收缩的导航栏.avi　　长度：4 分钟

17.6.1　实例描述

大家一定见过有些网站上的导航栏是可收缩的，这样既可以节省空间，又可使页面显示的内容更丰富一些。但这个功能具体是如何实现的呢？本节实例将演示它的具体实现过程。

17.6.2　实例应用

【例 17-6】会收缩的导航栏

(1) 新建一个 HTML 文件 navigation.html，在该文件中定义一个 div 元素，在该元素中设置 CSS、Java、Spring 等链接，形成一个导航菜单，代码如下所示：

```
<body>
    <div id="divMenu" style="position: absolute; top: 10; left: 40; width: 173;
     visibility: hidden; background-color:#CF9; height: 19"><nobr>
        <a href="http://www.cssmania.com/">CSS</a>-
        <a href="http://www.java.com/zh_CN/">Java</a>-
        <a href="http://www.springsource.org/">Spring</a>-
        <a href="http://www.hibernate.org/">Hibernate</a>-
        <a href="http://www.Java Script.com">Java Script</a>
        --- <a style="background-color: yellow; text-decoration: none"
        href="Java Script://" onclick="moveMenu()">技术导航菜单</a></nobr>
    </div>
</body>
```

(2) 使用 JavaScript 脚本声明定义了菜单移动的频率、移动速度等，代码如下所示：

```
<script language="Java Script">
    IE=document.all     <!--IE 的标志-->
    lshow=60
    var move=10;        <!-- 菜单移动的频率 -->
    menuSpeed=40        <!-- 菜单移动的速度为 40 秒一次 -->
    var moveOnScroll=true
    var tim;
    var ltop;
</script>
```

(3) 定义一个 b_getleft()函数用于获取菜单的左边界，代码如下所示：

```
    function b_getleft(){
        var gleft=eval(this.css.pixelLeft);        <!--获取菜单的左边界-->
        return gleft;
    }
```

(4) 定义 makeMenu()函数生成导航菜单，并设置菜单的状态、宽度等，调用 b_getleft()函数设置菜单左边界等，代码如下所示：

```
<script>
    function makeMenu(obj,nest){
        this.css=eval(obj+'.style')                <!--生成菜单-->
        this.state=1                               <!--设置菜单的状态-->
        this.go=0                                  <!--设置菜单的属性-->
        this.width=eval(obj+'.offsetWidth')        <!--设置菜单宽度-->
        this.left=b_getleft                        <!--设置菜单左边界-->
        this.obj = obj + "Object";
```

```
        eval(this.obj + "=this")
    }
</script>
```

(5) 定义收回菜单函数 mIn()，弹出菜单函数 mOut()，代码如下所示：

```
<script>
    function mIn(){                  <!--收回菜单-->
        if(oMenu.left()>-oMenu.width+lshow){
            oMenu.go=1               <!--设置标志位，表示菜单在运动中-->
            oMenu.css.left=oMenu.left()-move  <!--每次菜单的左边界向左移动一个值-->
            tim=setTimeout("mIn()",menuSpeed)  <!--设置移动的速度-->
        }else{
            oMenu.go=0               <!--设置标志位，表示菜单不在运动中-->
            oMenu.state=1            <!--设置标志位，表示菜单已经收回-->
        }
    }

    function mOut(){                 <!--弹出菜单-->
        if(oMenu.left()<0){
            oMenu.go=1               <!--设置标志位，表示菜单在运动中-->
            oMenu.css.left=oMenu.left()+move <!--每次菜单的左边界向右移动一个值-->
            tim=setTimeout("mOut()",menuSpeed)     <!--设置移动的速度-->
        }else{                       <!--菜单弹出完毕-->
            oMenu.go=0               <!--设置标志位，表示菜单不在运动中-->
            oMenu.state=0            <!--设置标志位，表示菜单已经弹出-->
        }
    }
```

(6) 定义 moveMenu()函数判断当前菜单的状态，如果为已弹出，清除定时器并将菜单收回，如果菜单未弹出，同样先清除定时器，再将菜单弹出，代码如下所示：

```
<script>
    function moveMenu(){
        if(!oMenu.state){            <!--如果菜单已经弹出-->
            clearTimeout(tim)        <!--清除定时器-->
            mIn()                    <!--将菜单收回-->
        }else{                       <!--如果菜单未弹出-->
            clearTimeout(tim)        <!--清除定时器-->
            mOut()                   <!--将菜单弹出-->
        }
    }
</script>
```

(7) 定义 menuInit()函数，调用 makeMenu()函数生成菜单，设置菜单的左边界，并将菜单设置为可视状态，代码如下所示：

```
<script>
    function menuInit(){
        oMenu=new makeMenu('divMenu')        <!--生成菜单-->
        oMenu.css.left=-oMenu.width+lshow    <!--设置菜单的左边界-->
        oMenu.css.visibility='visible'       <!--将菜单设置为可视-->
```

```
    }
</script>
```

(8) 定义在页面载入时调用 menuInit()函数，显示导航菜单栏，代码如下所示：

```
<script>
    onload=menuInit;          <!--页面载入时，直接调用函数 menuInit-->
</script>
```

17.6.3 运行结果

在 IE 浏览器中运行 navigation.html 文件，执行效果如图 17-7 所示。



图 17-7 会收缩的导航栏

17.6.4 实例分析



源码解析：

本实例首先在页面中定义了一个 div 元素，并在该元素中定义了几个链接形成一个导航菜单。然后定义菜单移动的频率、速度等变量，通过定义 makeMenu()函数生成导航菜单，并调用 b_getleft()函数设置菜单左边界，还定义了 moveMenu()函数，该函数通过调用 mIn()函数收回菜单，调用 mOut()函数弹出菜单。最后定义在页面载入时执行 menuInit()函数，生成菜单导航栏，并将菜单栏设置为可视，鼠标单击该菜单栏会收缩。

17.7 逐渐变色的表格

如果需要在页面中展示一些数据，使用表格非常方便，但使用标准的 HTML 语言生成的表格十分简单，而且也不美观，而使用 CSS 也只能为表格添加一些基本的外观样式，本节将

结合使用 JavaScript 实现动态改变表格的颜色。

视频教学：光盘/视频/17/17.7　逐渐变色的表格.avi　　长度：4 分钟

17.7.1　实例描述

最近在制作一个新闻管理网站，常常需要用到表格。看到其他网站的表格制作得一个比一个漂亮，自己也很想把网站做得漂亮一点，于是就开始研究表格特效，收益颇深。现在就来和大家分享一下我的收获，本节实例将演示如何使表格逐渐变色。

17.7.2　实例应用

【例 17-7】逐渐变色的表格

(1) 新建一个 table_color.html 文件，在该文件中定义一个银行卡信息表，包含卡号、密码、说明列字段，代码如下所示：

```
<table cellspacing=2 cellpadding=0 bgcolor=#999999 width=400>
   <caption>银行卡信息表</caption>
   <tr>
      <td width=33% align=center id=chcol0 name=id=chcol0>
       <font color=white>卡号
      </td>
      <td width=33% align=center id=chcol1 name=id=chcol0>
       <font color=white>密码
      </td>
      <td width=33% align=center id=chcol2 name=id=chcol0>
       <font color=white>说明
      </td>
   </tr>

   <tr>
      <td width=33% align=center id=chcol3 name=id=chcol0>
       <font color=white>10001
      </td>
      <td width=33% align=center id=chcol4 name=id=chcol0>
       <font color=white>password
      </td>
      <td width=33% align=center id=chcol5 name=id=chcol0>
       <font color=white>工商卡
      </td>
   </tr>
   <tr>
      <td width=33% align=center id=chcol6 name=id=chcol0>
       <font color=white>10002
      </td>
      <td width=33% align=center id=chcol7 name=id=chcol0>
```

```
         <font color=white>1978566
        </td>
        <td width=33% align=center id=chcol8 name=id=chcol0>
         <font color=white>交行卡
        </td>
    </tr>
     <tr>
        <td width=33% align=center id=chcol9 name=id=chcol0>
         <font color=white>10003
        </td>
        <td width=33% align=center id=chcol10 name=id=chcol0>
         <font color=white>886688
        </td>
        <td width=33% align=center id=chcol11 name=id=chcol0>
         <font color=white>中信卡
        </td>
    </tr>
</table>
```

(2) 设置表格的背景颜色及颜色的变化(采用十六进制数)、变化步长等，代码如下所示：

```
<script language="javascript">
    IE = (document.all);              <!--判断是否是 IE 浏览器-->
    if (IE) {
        var chidc = new Array();     <!--各个表格的背景色-->
        var hexc = new
Array('0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F');
<!--用于十一十六进制的转换-->
        var chidn = new Array(100,70,40,40,40,40,40,40,40,40,40,40);
        <!--初始化-->
        var step = new Array(10,10,10,10,10,10,10,10,10,10,10,10);
        <!--变化步长-->
        var chway = new
Array(step[0],step[1],step[2],step[3],step[4],step[5],
        step[6],step[7],step[8],step[9],step[10],step[11]);  <!--变化步长-->
        var tone = new Array(1,1,1,1,1,1,1,1,1,1,1,1);       <!--平衡变量-->
    }
</script>
```

(3) 定义一个 chcol()函数，在该函数中通过 for 循环依次改变表格的颜色，再设置 td 元素的 backgroundColor 背景色。代码如下所示：

```
<script>
    function chcol() {
        for (i=0; i<12; i++) {
            chidn[i]+=chway[i];  <!--三个表格分别按照设定的步长增长，最大值不能超过 255-->
            if (chidn[i]>=255) {
                chidn[i] = 255;
                chway[i] = -step[i];    <!--如果超过了，则改为同样步长的递减-->
            }
```

```
            else if (chidn[i]<=40) {                    <!--最小值不能低于 40-->
                chidn[i] = 40;
                chway[i] = step[i];
                tone[i]>=3? tone[i] = 1:tone[i]++; <!--tone 数组中的值不超过 3-->
            }
            col1 = hexc[Math.floor(chidn[i]/16)];  <!--转换为十六进制表示-->
            col2 = hexc[chidn[i]%16];
            tored = '';
            toblue = '';
            for (j=1; j<tone[i]; j++)
                tored+='00';    <!--分别对红色分量和蓝色分量进行处理以实现渐变效果-->
            for (j=3; j>tone[i]; j--)
                toblue+='00';
            chidc[i] = '#'+tored+col1+col2+toblue; <!--将值赋给 chidc-->
            td = eval('document.all.chcol'+i);
            td.style.backgroundColor = chidc[i];   <!--设定第 i 个表格的背景色-->
        }
        setTimeout('chcol()',100);                 <!--延时 0.1s-->
    }
</script>
```

(4) 使用 JavaScript 脚本调用 chcol()函数来改变表格的背景色，代码如下所示：

```
<script language="Java Script">
    chcol();  <!--调用 chcol 函数-->
</script>
```

17.7.3　运行结果

在 IE 浏览器中运行 table_color.html 文件，可以看到一个变色的表格，执行效果如图 17-8 所示。



图 17-8　变色的表格

17.7.4　实例分析



源码解析：

在本实例中首先在页面中定义一个银行卡信息表，存储一些银行卡的基本信息，然后声明了一些表格背景色，并使用十六进制的数据来标识颜色变化，最后定义了一个用来改变表格背景色的函数 chcol()。

17.8　常见问题解答

17.8.1　JavaScript 的典型错误



JavaScript 的一个错误，求解？
网络课堂：http://bbs.itzcn.com/thread-12397-1-1.html

我写了一个 JavaScript 程序，运行时报如下错误：

```
invalid label
[Break on this error] $('TipBox').style.display = (show==true):"","none";
```

$('TipBox') 有值，但是为什么使用三元运算符之后不能将正确的值赋值给 $('TipBox').style.display 呢？

【解决办法】你写的三元运算符格式有误，三元运算符的格式如下：

```
表达式？值 1：值 2
```

show=true 可以直接写成 show，将上面的代码改为下面的代码：

```
$('TipBox').style.display = show?"":"none";
```

或者

```
$('TipBox').style.display = show?"block":"none";
```

17.8.2　变量每秒自加 1



在 JavaScript 中如何使变量每秒自加 1？
网络课堂：http://bbs.itzcn.com/thread-12398-1-1.html

本人初次接触 JavaScript，想实现用 JavaScript 写的变量每秒加 1，当达到一定数值时弹出窗口显示该数值，但由于使用 JavaScript 比较生疏，希望高手能给出相应的实现代码。

【解决办法】使变量每秒加 1，代码如下：

```
var count = 0;
window.setInterval("showNumber()", 1000); // 毫秒单位
function showNumber() {
    count++;
    if (count == 100) {
        alert(count);
    }
}
```

17.8.3　通过滤镜实现图片翻转的问题



通过滤镜实现图片翻转的问题。

网络课堂：http://bbs.itzcn.com/thread-12399-1-1.html

有一张图片，它控制一个 div 的 display 属性。如何实现单击图片改变自身的翻转方向呢？编写了下面一段 JavaScript 代码：

```
function bian() {
    if (document.getElementById("img1").style.filter == "fliph") {
        document.getElementById("img1").style.filter = "FlipH";
    }
    else {
        document.getElementById("img1").style.filter = "FlipV";
    }
}
```

上述可以解决垂直翻转的问题，而我需要的是水平的，另外，上面的垂直翻转会停在翻转处不再循环。

style.filter = "FlipH"用于设置图片水平翻转。应用上段代码之后，第一次单击可以实现图片的水平翻转，但是之后单击图片不会动，我需要实现该张图片能连续翻转的特效。怎么实现？

【解决办法】用新的滤镜 BasicImage，简单的使用方法实例如下：

```
<DIV ID="oFilterDIV" STYLE="position:absolute; top:50px; left:10px;
width:240px;
height:160px; padding:10px; font:bold 13pt verdana; background: yellowgreen;
filter:progid:DXImageTransform.Microsoft.BasicImage(grayscale=0, xray=0,
mirror=0,
 invert=0, opacity=1, rotation=0)">
This is the DIV content.</DIV>
```

我们要关注的主要是 rotation 和 mirror 两个参数的使用。rotation 就是让图片顺时针转动，取值 0-3，转动的度数为 90 的倍数即可。mirror 就简单了，也就是让图片以镜像方式显示。相信通过设置这两个参数就可以达到任意翻转图片的效果了。

17.9　习　　题

1. 填空题

(1) JavaScript 的数据类型主要有字符串值、整数、浮点数、逻辑值和____________。
(2) JavaScript 运算符包括== 、+=、%= 、++和____________。
(3) 下面代码运行的结果是____________。

```
function hi(){
    var a;
    alert(a);
}
```

2. 选择题

(1) 以下使用 JavaScript 生成对象的方法中，正确的是____________。

A. var z = new Boolean(a);　　B. var str = Java Script;
C. fruit=new Array;　　D. today1=new Date(2008,10,1);

(2) 有关 JavaScript 语句，下列说法正确的是____________。

A. 单行注释语句是在需要注释的行前面用\\
B. with 语句的功能是为一段程序建立默认对象
C. JavaScript 中没有 if...else 语句
D. JavaScript 中只有 while 语句，而没有 do...while 语句

(3) 求一个表达式的值，可以使用的函数有____________。

A. eval()　　B. isNaN()　　C. parseInt()
D. parseFloat()　　E. return()

(4) JavaScript 的 onSubmit 事件的作用是____________。

A. 当一个表单中的对象被单击时，执行 JavaScript 事件
B. 当用户提交一个表单时，需要执行 JavaScript 事件
C. 当鼠标移出对象时发生的事件
D. 对象发生改变时调用的事件
E. 当一个鼠标指针从对象或区域之间移到对象或区域上时，调用的事件

3. 上机练习

上机练习：闪烁变色的链接

练习要求：本次练习首先需要定义闪烁的颜色变化顺序，然后定义变化时的颜色数组。每种颜色闪烁的时间，最后定义一个为链接和下划线赋颜色取值的函数，并通过 setTimeout()函数，按每种颜色闪烁的时间执行一次为链接和下划线赋颜色取值的函数，执行效果如图 17-9 所示。



图 17-9　闪烁变色的链接



第 18 章　CSS 跨浏览器支持技术

内容摘要：

经常有朋友问我，网站要在哪些浏览器上测试？要达到怎样的兼容？我想这或许不仅仅是他们的问题，也许是每个 Web 设计人员的疑问。

由于 CSS 标准的出现比浏览器要晚，所以当时的浏览器还不能完全支持 CSS 样式表，只是支持 CSS 样式表中的部分规则。直到 CSS 2.0 版本的出现，才被广大浏览器厂商和网页设计师所重视和接受。

由于各浏览器在遵循 CSS 标准的同时又进行了自己的扩展，因此就会出现 CSS 样式在不同浏览器上显示的效果不一致的情况。

本章以目前主流浏览器 IE6、IE8、Firefox 和 chrome 为例，首先介绍 4 个典型的 CSS 兼容性问题的解决办法，然后对 Web 设计人员如何在开发中避免 CSS Bug 的问题进行了讨论，最后提供了一些编写 CSS 兼容性的建议。

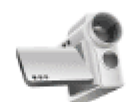
学习目标：

- 掌握 CSS 条件注释的用法
- 掌握空白边叠加的解决办法
- 了解浮动元素产生闭合的原因
- 熟悉一种闭合浮动的解决办法
- 了解 DOCTYPE 的作用
- 熟悉各个 DOCTYPE 的定义方法
- 了解校验 CSS Bug 的方法
- 熟悉捕捉 CSS Bug 的基本步骤
- 了解通用的 CSS 编写建议

18.1　为特定浏览器提供不同的样式

目前，对于 CSS 开发人员来说最大的问题是绝大多数用户仍然在使用 IE6，而它是一个 CSS 规范支持较差，问题较多的浏览器。虽然微软在 IE6 的更高版本中修复了很多众所周知的缺陷，并添加了更多对 CSS 2.0 规范的支持，但仍然有许多用户无法或者不想从 IE6 升级到其他浏览器。

利用 IE 的条件注释可以很好地解决这个问题，针对特定的浏览器提供不同的样式。

 **视频教学：** 盘/视频/18/18.1　给特定浏览器提供不同的样式.avi　　长度：5 分钟

18.1.1　基础知识——条件注释

条件注释(Conditional Comments)是一种安全地区分 IE 浏览器版本的语法，且被认为是取代针对 IE CSS hack 的首选办法。

其实条件注释就是一些 if 判断，但这些判断不是在脚本里执行的，而是直接在 HTML 代码里执行的。例如，如下是一个简单实例：

```
<!--[if IE]>
这里是正常的 html 代码
<![endif]-->
```

从这个最简单的实例中，我们可以看到条件注释具有如下特性。

- 条件注释的基本结构和 HTML 的注释(<!-- -->)是一样的。因此 IE 以外的浏览器将会把它们看做是普通的注释而完全忽略它们。
- IE 将会根据 if 条件来判断是否如解析普通的页面内容一样解析条件注释里的内容。
- 条件注释使用的是 HTML 的注释结构，因此它们只能用在 HTML 文件里，而不能在 CSS 文件中使用。

根据上面的介绍，对其进行扩展便可以使用如下代码检测当前 IE 浏览器的版本(注意：在非 IE 浏览器中是看不到效果的)。

```
<!--[if IE]>
     <h1>您正在使用 IE 浏览器</h1>
     <!--[if IE 5]>
         <h2>版本 5</h2>
     <![endif]-->
     <!--[if IE 5.0]>
         <h2>版本 5.0</h2>
     <![endif]-->
     <!--[if IE 5.5]>
         <h2>版本 5.5</h2>
     <![endif]-->
     <!--[if IE 6]>
         <h2>版本 6</h2>
```

```
    <![endif]-->
    <!--[if IE 7]>
       <h2>版本 7</h2>
    <![endif]-->
<![endif]-->
```

如果当前的浏览器是 IE，但版本比 IE5 还低，该怎么办呢，可以使用“<!--[if ls IE 5]>” 条件注释。但由于条件注释只能在 IE5+的环境之下，所以“<!--[if ls IE 5]>”根本不会被执行。此时，便可以使用条件注释中的表达式，它有如下可选值。

- lte：就是 Less than or equal to 的简写，也就是小于或等于的意思。
- lt：就是 Less than 的简写，也就是小于的意思。
- gte：就是 Greater than or equal to 的简写，也就是大于或等于的意思。
- gt：就是 Greater than 的简写，也就是大于的意思。
- !：就是不等于的意思，跟 Java Script 里的不等于判断符相同。

例如，如下是应用这些表达式的实例语句：

```
<!--[if gt IE 5.5]>          / 如果 IE 版本大于 5.5 /
<!--[if lte IE 6]>           / 如果 IE 版本小于等于 6 /
<!--[if !IE]>                / 如果浏览器不是 IE /
```

本节开始就说明了，由于 IE 各版本的浏览器对制作的 Web 标准的页面解释不一样，具体就是对 CSS 的解释不同，因此为了使不同版本的浏览器都能够兼容，可运用条件注释来各自定义，最终达到兼容的目的。例如：

```
<!-- 默认先调用 css.css 样式表 -->
<link rel="stylesheet" type="text/css" href="css.css" />

<!--[if IE 7]>
<!-- 如果 IE 浏览器版本是 7，调用 ie7.css 样式表 -->
<link rel="stylesheet" type="text/css" href="ie7.css" />
<![endif]-->

<!--[if lte IE 6]>
<!-- 如果 IE 浏览器版本小于等于 6，调用 ie.css 样式表 -->
<link rel="stylesheet" type="text/css" href="ie.css" />
<![endif]-->
```

这样就能够区分 IE7 和 IE6 以下的浏览器对 CSS 的执行，从而达到兼容的目的。同时，首行默认的 css.css 还能与其他非 IE 浏览器实现兼容。



默认的 CSS 样式应该位于 HTML 文档的首行，进行条件注释判断的所有内容必须位于该默认样式之后。

例如如下代码，在 IE 浏览器下执行显示为红色，而在非 IE 浏览器下显示为黑色。如果把条件注释判断放在首行，则不能实现。该实例就能够说明如何解决网页对 IE 浏览器和非 IE 浏览器的兼容性问题。

```
<style type="text/css">
body{
```

```
background-color: #000;
}
</style>
<!--[if IE]>
<style type="text/css">
body{
background-color: #F00;
}
</style>
<![endif]-->
```

在上述代码中，使用“<!--[if !IE]>”代码来定义非 IE 浏览器下的样式。但是在 IE 5 及其以上版本中，<!--[if ls IE 5]>表达式将不会被执行，而是当作注释被忽略。

18.1.2 实例描述

在浏览器厂商市场分割及版本多元化的今天，单单是给 CSS 做优化已不能满足页面制作的需要。完美的页面结构，可以省去很多的优化步骤，但对于版本间不同的盒子模型来说。与其写 hack，还不如为每个不同解释定义的版本写一套 CSS。无论在制作时间上还是维护成本上，这都是一个可以考虑的解决方案。

这就用到了 CSS 的条件注释，这是一种只在 Windows IE 上被识别的注释，从 IE5 开始支持。这就是说，被定义于其中的东西只在 IE 下被识别，这就成为一个能够很好的跨浏览器样式一致性问题的解决方案。

18.1.3 实例应用

【例 18-1】给特定浏览器提供不同的样式

(1) 打开一个需要针对浏览器来设置样式的 HTML 文件。然后，在能够正常显示该页面的 IE 浏览器中打开，查看效果。如图 18-1 所示，为本实例在 IE6 中显示正常的页面效果。



图 18-1 IE6 中显示的效果

(2) 使用更高版本的 IE 浏览器打开同一个页面，查看效果是否相同。如图 18-2 所示为 IE8 中的运行效果。

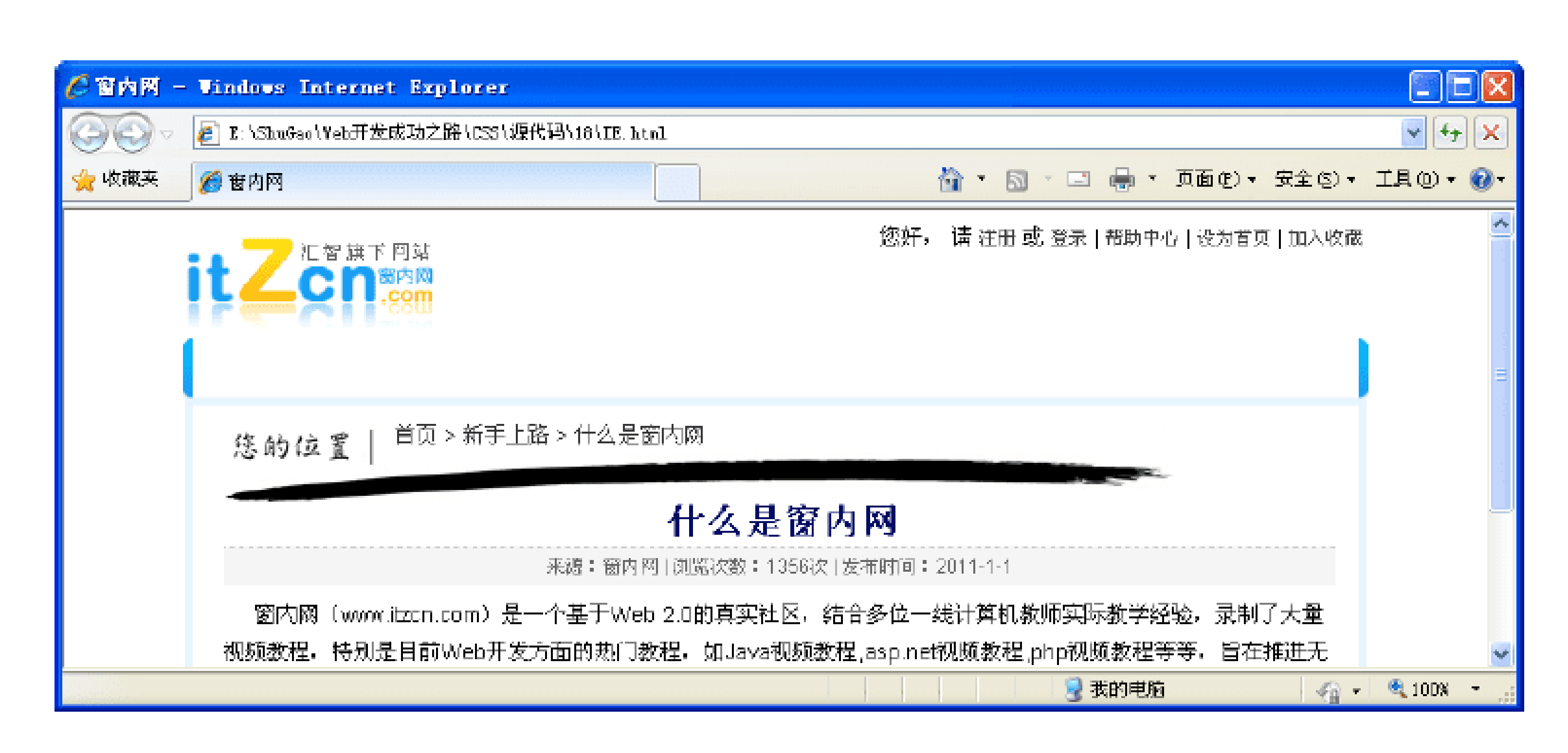


图 18-2　IE8 中显示的效果

(3) 对比以上两图就会发现，在 IE8 中页面的导航菜单显示不正常。这时便需要通过条件注释来针对 IE8 浏览器修复这个问题。如下所示是这个菜单的 HTML 代码：

```
 <DIV id="nav_index">
   <ul>
     <LI><A href="http://school.itzcn.com">教育学院</A> </LI>
     <LI><A href="http://bbs.itzcn.com">学术论坛</A> </LI>
     <LI><A href="http://blog.itzcn.com">技术文档</A> </LI>
     <LI><A href="http://home.itzcn.com">娱乐空间</A> </LI>
     <LI><A href="http://www.itzcn.com">窗内首页</A> </LI>
     <LI><SPAN></SPAN></LI>
   </ul>
 </DIV>
```

(4) 在页面的 head 元素内链接外部 CSS 的下方，添加如下的条件注释语句针对 IE8 来单独设置 CSS 样式文件。

```
<!--[if IE 8]>
  <LINK href="ie8.css" type="text/css" rel="stylesheet">
<![endif]-->
```

(5) 打开 ie8.css 文件，然后根据上步给出的效果和 HTML 代码重复编写 CSS 样式，最终达到相同的效果。代码如下所示：

```
#nav_index {
    MARGIN:5px 0px;
    WIDTH:762px;
    height:38px;
    POSITION:relative;
    background-image: url(images/navbg.gif);
    background-repeat: repeat;
    background-position: 0px -76px;
    clear:both;
}
```

(6) 保存对 HTML 页面的修改，完成实例的制作。

18.1.4 运行结果

由于条件注释只针对 IE 有效，所以现在只需要在 IE8 中打开看看是否显示正常就可以了。运行效果如图 18-3 所示，显示正常，说明条件注释起作用了。



图 18-3 IE8 中显示的正常效果

18.1.5 实例分析



源码解析：

本实例中由于原来的 HTML 页面在 IE8 中显示不正常，所以利用 CSS 条件注释对匹配的 IE8 浏览器单独编写 CSS 样式，以达到显示正常的效果。

CSS 条件注释的优势非常明显，而且也非常容易记住，是对常规 HTML 注释的扩展。主要的缺点就是这些注释需要放在 HTML 中，而不是放在 CSS 中。这样一来，当我们希望不再支持某种浏览器时，就需要在每个 HTML 页面中删除针对该浏览器的 CSS 条件注释。

18.2 解决空白边叠加问题

即使一个有经验的 Web 设计人员在制作页面时，也难免会出现空白边叠加的问题。

出现这种问题的原因并不是浏览器的解释有 Bug，或者浏览器对 W3C 的支持不好，而是由于 W3C CSS2.1 规定的正常现象。

为什么在 IE 浏览器中没有呢？这是因为在 IE 浏览器中设定了宽度和高度，触发了 IE 的布局事件，如果去除例子中的宽度和高度(记得给子元素增加内容文字)，就会发现它的实际效果其实和开始在 Firefox 中是一致的(也就是说，出现这种现象是正常的)。

视频教学：光盘/视频/18/18.2 解决空白边叠加问题.avi 长度：5 分钟

18.2.1　基础知识——空白边叠加

简单地说，当两个垂直的空白边相遇时，它们将会形成一个空白边。这个空白边的高度等于两个发生叠加的空白边中的较大者。

当一个元素出现在另一个元素上面时，第一个元素的低空白边与第二个元素的顶空白边发生叠加。例如，下面的 CSS 样式代码：

```
#Box1 {
    height: 40px;
    border: 1px solid #000;
    margin: 30px;
    background-color: #CCC;
}
#Box2 {
    margin: 30px;
    border: 1px solid #000;
    background-color: #CCC;
    height: 40px;
}
```

两个元素的空白边叠加的效果如图 18-4 所示。

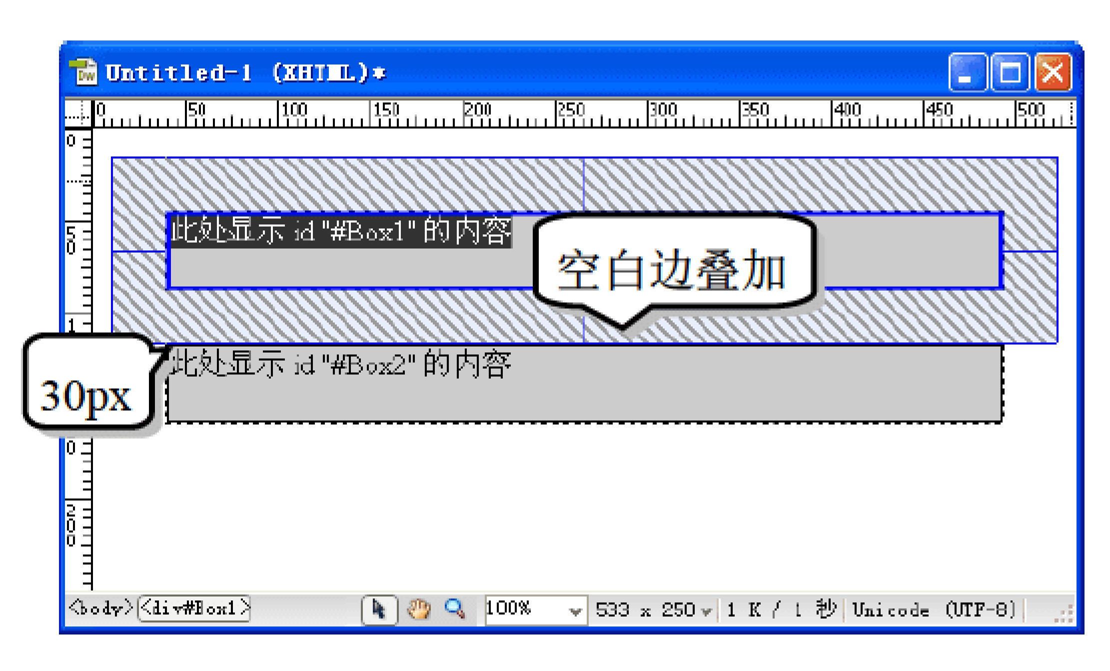


图 18-4　空白边叠加

当一个元素包含在另一个元素中时(假设没有填充或边框将空白边分隔开)，它们的顶和底空白边也会发生叠加。还有一种情况，假设有一个空元素，它有空白边，但是没有边框和填充。在这种情况下，顶空白边与底空白边就碰到了一起，此时也会发生叠加。

空白边叠加初看上去可能有点奇怪，但是它实际上是有意义的。例如，以由多个段落组成的典型文本页面为例。第一个段落上面的空间等于段落的顶空白边。如果没有空白边叠加，那么后续所有段落之间的空白边将是相邻顶空白边和底空白边的和，这意味着段落之间的空间是页面顶部的两倍。如果发生空白边叠加，段落之间的顶空白边和底空白边就叠加在一起，这样各处的距离就一致了，如图 18-5 所示。



只有普通文档流中区块的垂直空白边才会发生叠加。对于行内块、浮动块或绝对定位块之间的空白边是不会叠加的。

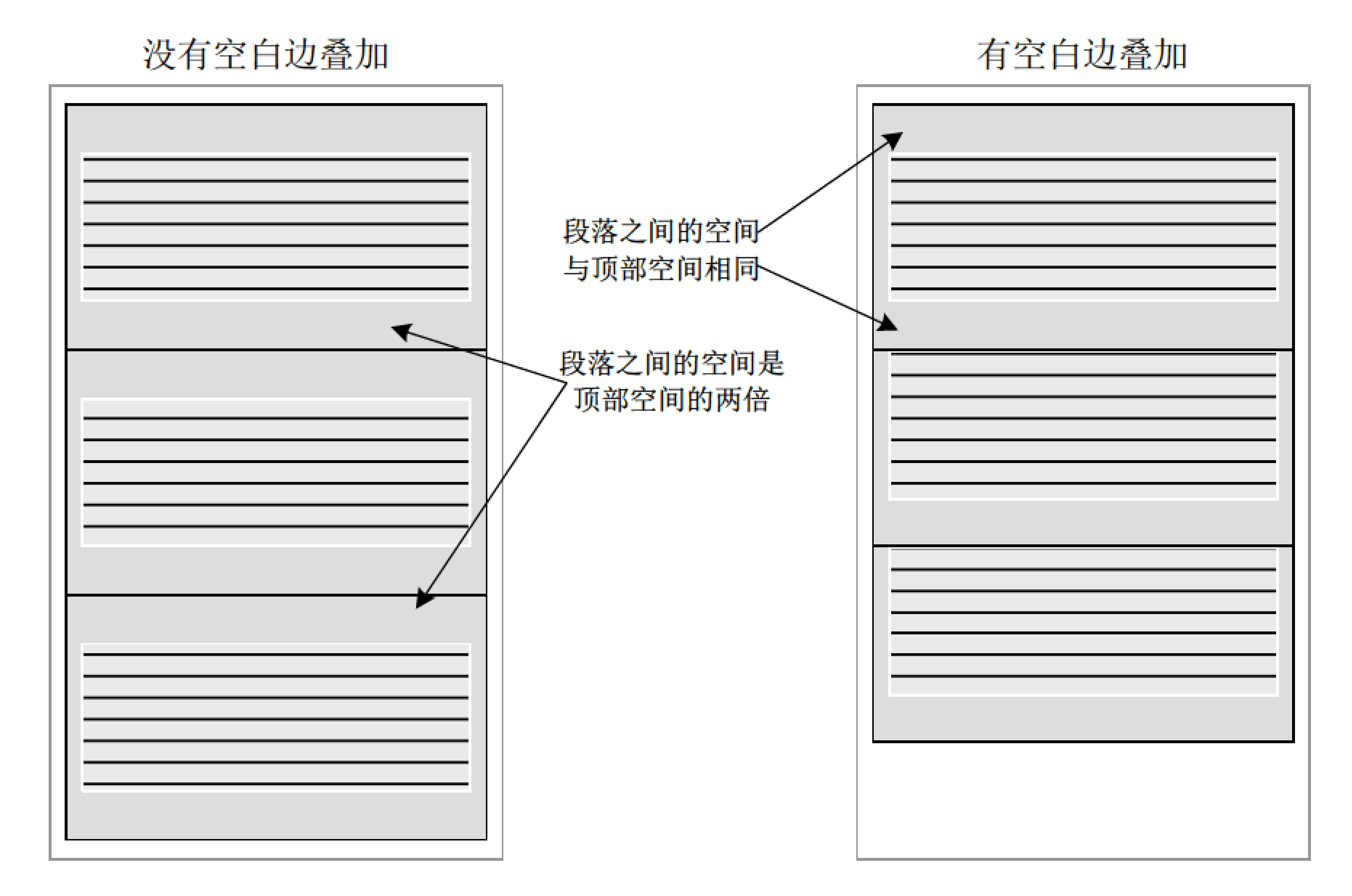


图 18-5　维护段落间距的空白边

18.2.2　实例描述

空白边叠加是一个相当简单的概念，但是在实际对网页进行布局时，它会造成许多混淆。很多时候我们并不希望这样，下面是一些常用的解决方法。

18.2.3　实例应用

【例 18-2】 解决空白边叠加问题

(1) 新建一个 HTML 页面，命名为 collapsingMargin.html。

(2) 在页面的 body 元素中添加如下的代码：

```
<div id="wrap">
  <div id="header" >经典语录</div>
  <div id="content">
    <ul>
      <li>窗外放晴了，屋内仍继续下雨。我微笑，并不等于我快乐。我撑伞，并非只是为了避雨。你永远都不懂我在想什么。</li>
      <li>我努力寻找希望，生怕幸运就在身边，却被我粗心错过。</li>
      <li>飞起来的时候，幸福的感觉就来了。 </li>
      <li>风吹过湖面，柳条摇曳，石头上布满苍老的青苔，湖底铺着陈腐的枯叶，鱼儿在清澈水里相互追逐，小鸟在鸣唱，空气中弥漫着果香，这里仿若仙境。
        而我心中却只是在担心，回家时该如何告诉妈妈，我又不小心掉落了一双心爱的小红鞋</li>
      <li>我的心开始下雪，雪无声地覆盖了所有，湮灭了迷惘，骄傲与哀痛，当一切归于寂静时，世界突然变得清亮明朗。
        所以，别为我忧伤，我有我的美丽，它正要开始…… </li>
    </ul>
```

```
  </div>
</div>
```

(3) 为页面设置一个背景图片，然后定义#wrap 容器在页面中的显示样式。

```
#wrap {
    text-align: left;
    margin: 130px 30px 30px 30px;
    padding: 20px;
    width: 500px;
    border: 5px solid #6F3623;
    overflow: auto;
}
```

(4) 定义#header 中标题的显示样式，并设置 margin 属性，代码如下：

```
#header {
    margin:10px;
    color:#006;
    font-size:24px;
    font-weight:bold;
    text-align:center;
    background-color:#EAEAEA;
    border-bottom: 3px dashed #703724;
}
```

(5) 为#content 中的内容定义样式，也包含 margin 属性，代码如下：

```
#content {
    margin:10px;
    border:1px solid red;
}
```

(6) 在浏览器中打开查看效果。由于#header 和#content 位于同一层次，且都设置了 margin 属性，此时会发现空白边叠加效果导致的间隙比较小，如图 18-6 所示。



图 18-6　空白边叠加时的运行效果

(7) 要解决这个问题，首先在#content 的样式中添加“float:left”属性，然后在#header 和#content 容器之间添加如下代码：

```
<div style="clear:both"></div>
```

(8) 保存对 HTML 页面的修改，完成实例的制作。

18.2.4 运行结果

再次运行，在浏览器中查看运行效果。此时会看到，#header 和#content 之间的间距变成了双倍，即 20px，如图 18-7 所示。

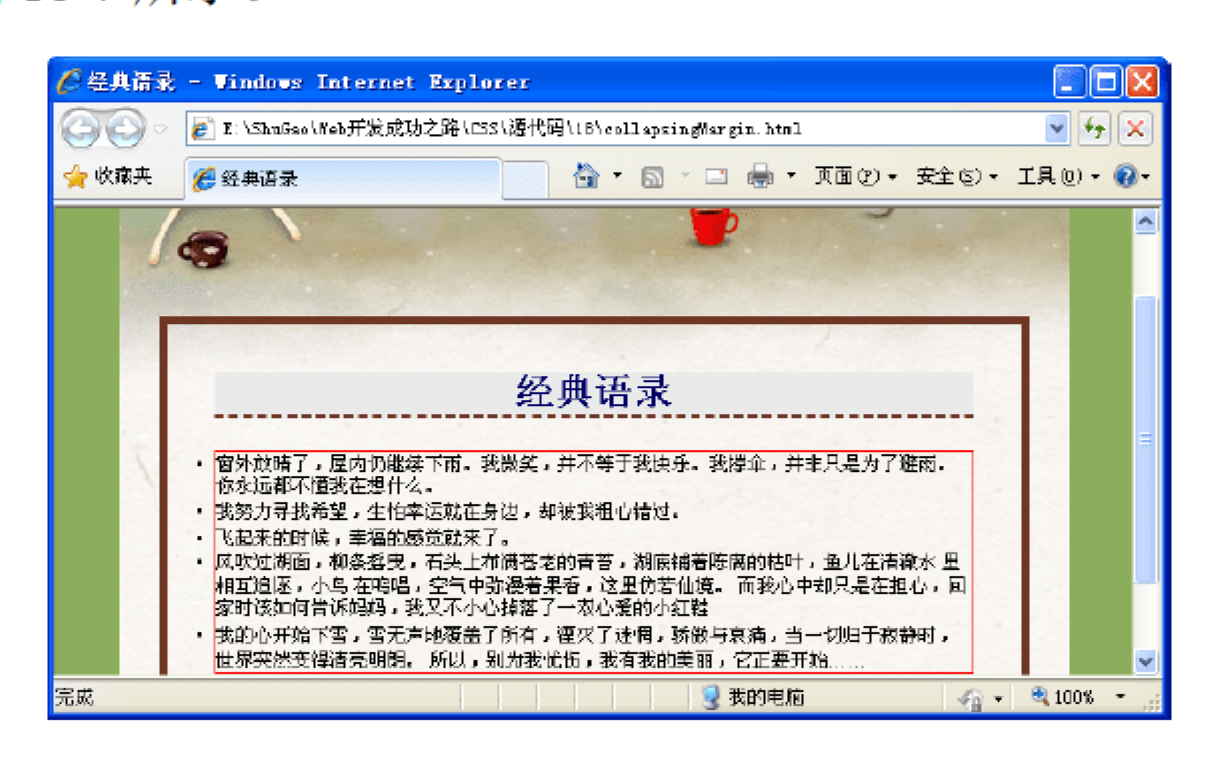


图 18-7 解决叠加后的运行效果

18.2.5 实例分析



源码解析：

通过本实例，我们了解了产生空白边叠加的原因，而且掌握了出现此问题时的解决办法。

但是要注意，有时空白边叠加问题并不需要解决，例如在显示大段文本或者段落时。因此，读者在实际设计页面时应该区分何时希望产生叠加，并加以处理。

18.3 避免浮动元素闭合

CSS 中的 float 属性在页面排版上非常有用，但同时也带来了很多问题。最常见的就是浮动元素的闭合问题。



视频教学：光盘/视频/18/18.3 避免浮动元素闭合.avi　　长度：6 分钟

18.3.1 基础知识——闭合浮动元素

很多人都研究过闭合浮动元素的问题，但是解决方法却不一样，也并不是每一种方法都尽善尽美。闭合浮动元素(或者叫清除浮动)是 Web 标准设计中经常会遇到的一个问题，因此，这

里总结一下目前经常用到的几种方法，并比较一下它们的易用性和适用环境。

1. 产生原因

最简单的一种情形就是我们把一个小的、固定宽度的 div 元素(比如导航、引用等)和其他元素内容一起包含在一个大的 div 中。比如下面这段代码：

```
<div id="outer">
  <div id="inner"> <h2>我的地盘 听我的</h2> </div>
  <h1>我的理想</h1>
  <p>春暖花开 面朝大海</p>
</div>
```

可以为 id 为“#inner”的层设定一个宽度值(比如说 20%)，但是由于 div 是块级元素，即使设定了宽度，其后面的内容也只能在下一行中显示，除非我们给它设定一个浮动属性(向左浮动或者向右浮动)。那么此时就会产生我们上面提到的问题了。

如果 id 为“#inner”的层的宽度和高度都比“#outer”小，就不会有问题。

但是，如果“#inner”的高度超过了“#outer”，那么它的底部就会超出 id 为“#outer”的层的底部。这是因为为 id 为“#inner”的层设定 float 属性后，它就会脱离文本流，无论其宽度和高度怎么变化都不会使 id 为“#outer”的层跟随变化。这就是我们所提到的“清除浮动(闭合浮动元素)”或者是“闭合浮动”问题。

2. 解决办法

通过前面的描述，在了解闭合浮动元素产生的原因之后，解决起来自然就不费事了。目前针对这个问题，主要有 4 种解决办法，下面依次介绍这几种方法。

1) 增加标签法

这也是 W3C 推荐的方法，就是在内容的末尾增加一个空的标签。典型的做法就是使用类似下面这样的代码：

```
<br style="clear:both;" />
```

或者：

```
<div style="clear:both;"></div>
```

这种办法通过增加一个 HTML 元素迫使外部容器撑开。不过这个办法会增加额外的标签，使 HTML 结构看起来不够简洁。

2) 使用 after 伪类

使用 after 伪类和内容声明在指定的末尾添加新的内容。经常的做法就是添加一个“点”，因为它比较小，不太引人注意。然后我们再利用它来清除浮动(闭合浮动元素)，并隐藏这个内容：

```
#outer:after{
   content:".";
   height:0;
   visibility:hidden;
   display:block;
   clear:both;
  }
```

但奇怪的是，Windows 中的 IE6 及以下版本并不支持 CSS 2.0 中的 after 伪类，而 Mac 中的 IE 却可以正常使用，所以我们还要单独针对 Win/IE 进行处理。在区分 Win 和 Mac 中 Internet Explorer 的诸多方法中，最常用就是 Holly 招数。Holly 招数的原理是这样的，其 CSS 代码如下：

```
/* 这段代码只有 Win/IE 可以看见 \*/
CSS 规则
/* 结束 Hack */
```

上面的代码中有两行注释，其中第一行结束时出现了反斜杠(\)，在 Mac 的 IE 中会认为注释并没有结束，于是继续向下执行直到第一个结束的“*/”出现，这中间的所有字符都被当作是注释，而 Win/IE 却也只会把第一行和第三行看做是注释，中间的为有效代码。所以这样就区分出来了不同平台上的 IE 了。

正是基于以上原理，在 Windows 的 IE 6 中清理浮动，可以使用如下代码：

```
#outer {
   display:inline-block;
}
/* Holly Hack Begine \*/
* html #outer {
   height:1%;
}
#outer {
   display:block;
}
/* End Hack */
```

例如，如果不考虑 Mac/IE6 用户，只需要写“* html #outer {height:1%;}”即可。另外，似乎在 IE7 中“display:inline-block”并不起作用。因此要使用最完整的 Hack 招数。



如果对如何使用 CSS 2 中的伪类不熟悉，可以参见本书的第 3 章。

3) 浮动外部元素(float-in-float)

这种方法很简单，就是把“#outer”元素也进行浮动(向左或者向右)。

但这种方法同时也会带来一个问题，就是和“#outer”相邻的下一个元素会受到“#outer”的影响，位置会产生变化，所以使用这种方法一定要小心。



有选择性地把页面中的所有元素都浮动起来，最后使用一个适当的、有意义的元素(比如页脚)进行清理浮动，这有助于减少不必要的标记。

4) 设置 overflow 为 hidden 或者 auto

把“#outer”样式的属性 overflow 值设置为 hidden 或者 auto 同样可以清理浮动，这种方法不需要添加额外的标记。但是使用 overflow 属性的时候一定要小心，因为它会影响浏览器的表现。

另外，在 IE6 中单纯地设置 overflow 为 hidden 或者 auto 并不能有效清除浮动(闭合浮动元

素)，还要指定“#outer”样式的一个维度，即要么给它指定一个宽度，要么指定一个高度：

```
#outer {
    overflow:auto;
    width:100%;
}
```



如果要在 Mac/IE5 上使用这种方法清除浮动(闭合浮动元素)，就必须设定 overflow 的属性值为 hidden。

5) 比较与选择

下面我们来比较一下这 4 种方法。首先，不推荐使用 after 伪类。对比其他三种方法，holly 招数有点太烦琐，不好掌握，而且这种方法存在更多的不确定性。推荐对代码有“洁癖”并且技术较高的编程人员使用。那么，其他三种方法其实都可以考虑。

- 不过使用 overflow 时，可能会对页面表现造成影响，而且这种影响是不确定的，最好能在多个浏览器上测试页面。
- 而对于使用额外标签清除浮动(闭合浮动元素)，是 W3C 推荐的做法。至于使用
元素还是空<div></div>可以根据自己的喜好来选(当然也可以使用其他块级元素)。不过要注意的是，
本身是有表现的，它会多出一个换行，所以要设定它的 height 为 0，以隐藏它的表现。在大多数情况下使用空<div>比较合适。
- float-in-float 也是一个很好的选择，把要进行清理浮动的元素外层再加上一层<div>，并设置属性 float:left 即可，不过要注意相邻元素的变化。

18.3.2　实例描述

最近两个月陆续做了几个网站的美工，做的过程中对 div+CSS 的标准开发有了进一步的了解。有两点收获最大，一是彻底弄懂了 CSS 的盒子模式(Box Model)，再一个就是搞定了困扰我很久的“闭合浮动元素”的问题。

一般说来如果某个子元素使用了浮动(float)，那么父元素总是不能确切地知道子元素是在什么位置结束的，所以父元素的下边框总是从子元素的中间甚至是顶部穿过，看起来很不舒服。

18.3.3　实例应用

【例 18-3】避免浮动元素闭合

(1) 新建一个 HTML 页面，命名为 ClearFloat.html。

(2) 在页面的 body 元素中添加如下代码：

```
<div id="wrap">
  <div class="left">
    <h2>摇篮网小黑板</h2>
     <!-- 此处省略其他 HTML 代码 -->
  </div>
  <div class="right">
```

```
    <h2>怎么联系我们</h2>
     <!-- 此处省略其他 HTML 代码 -->
  </div>
</div>
```

(3) 为页面设置一个背景图片，然后定义#wrap 主容器在页面中的显示样式，代码如下：

```
#wrap {
    margin: 20px;
    padding: 10px;
    border: 5px solid #F0FCFC;
    text-align: left;
}
```

(4) 定义左边 div 容器的样式类 left，代码如下所示：

```
.left {
    float:left;
    width:38%;
    padding:10px;
}
```

(5) 定义左边 div 容器的样式类 right，代码如下所示：

```
.right {
    float:right;
    width:52%;
    padding:10px;
    border-left-width: 6px;
    border-left-style: solid;
    border-left-color: #6B9F4A;
}
```

(6) 在浏览器中打开查看效果。由于 left 和 right 位于同一层次，且都设置了 float 属性。此时会发现主容器并没有随子容器的内容而扩展，如图 18-8 所示。



图 18-8　未闭合浮动元素的运行效果

(7) 根据前面介绍过的内容，解决办法非常简单，只需为主容器#wrap 添加“overflow:auto;”代码即可。

(8) 保存对 HTML 页面的修改，完成实例的制作。

18.3.4　运行结果

在浏览器中运行，会看到闭合浮动元素的运行效果，如图 18-9 所示。



图 18-9　闭合浮动元素的运行效果

18.3.5　实例分析



源码解析：

对比图 18-8 和图 18-9 就可以看出，在为主容器#wrap 添加“overflow:auto;”代码之后，它的高度随子容器的内容而向下扩展，从而清除了浮动元素的闭合问题。

当然，我们也可以在容器之后增加如下的空标签来解决这个问题。

```
<div style="clear:both"></div>
```

18.4　选择合适的 DOCTYPE

DOCTYPE 声明位于文档中最前面的位置，处于<html>标签之前。此标签可告知浏览器，文档使用哪个版本的 HTML 或 XHTML 规范。

该标签可声明三种 DTD 类型，分别表示严格版本、过渡版本以及基于框架的 HTML 文档。



视频教学：光盘/视频/18/18.4　选择合适的 DOCTYPE.avi　　　　长度：11 分钟

18.4.1　基础知识——DOCTYPE 的作用

除了 IE 本身的缺陷之外，最可能的问题是因为 IE 在显示页面时使用了“怪异”模式。许多现代浏览器都提供了两种显示模式。“怪异”模式在显示文档时，使用的是像 Netscape 4、IE5 等老版本浏览器中具有较多缺陷的版本。标准模式则是按照 W3C 规范来显示文档的。

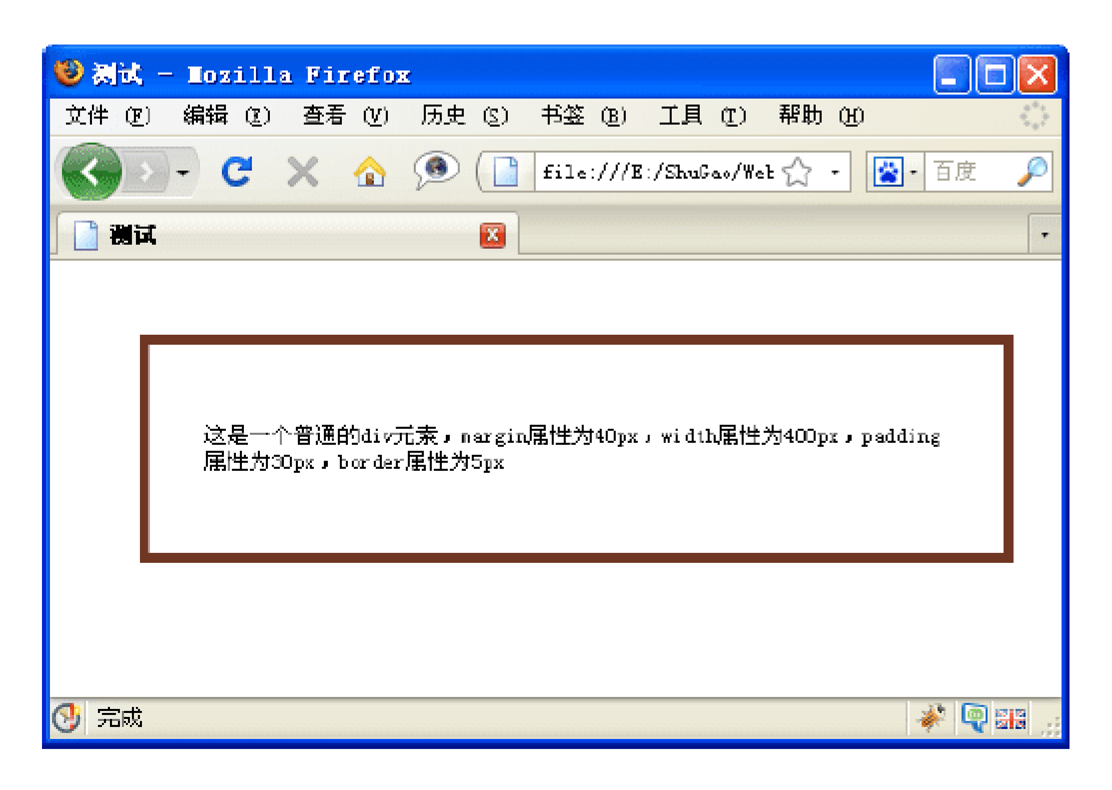
- 使用老版本 DOCTYPE 的文档通常在格式化方面存在不足，或者那些根本没有使用 DOCTYPE 的文档，会直接使用“怪异”模式显示。
- 使用严格的 XHTML DOCTYPE 的文档，就会以标准模式显示。

但是仍然需要注意，这对于 IE 6 来说并不是很简单的问题。例如，不管在文档最前面的 DOCTYPE 语言中指定什么，包括 XML 声明，IE 6 仍然会以“怪异”模式来显示。

例如，有一个 XHTML 文档，里面包含了如下内容：

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>测试</title>
</head>
<style type="text/css">
#wrap {
    margin: 20px;
    width: 300px;
    padding: 10px;
    border: 5px solid #6F3623;
    overflow: auto;
    text-align: left;
}
</style>
<body>
<div id="wrap">
  <p>这是一个普通的 div 元素，margin 属性为 20px，width 属性为 300px，padding 属性为 10px，
border 属性为 5px </p>
</div>
</body>
</html>
```

如图 18-10 所示为这个文档在 Firefox 与 IE 6 中的显示效果。



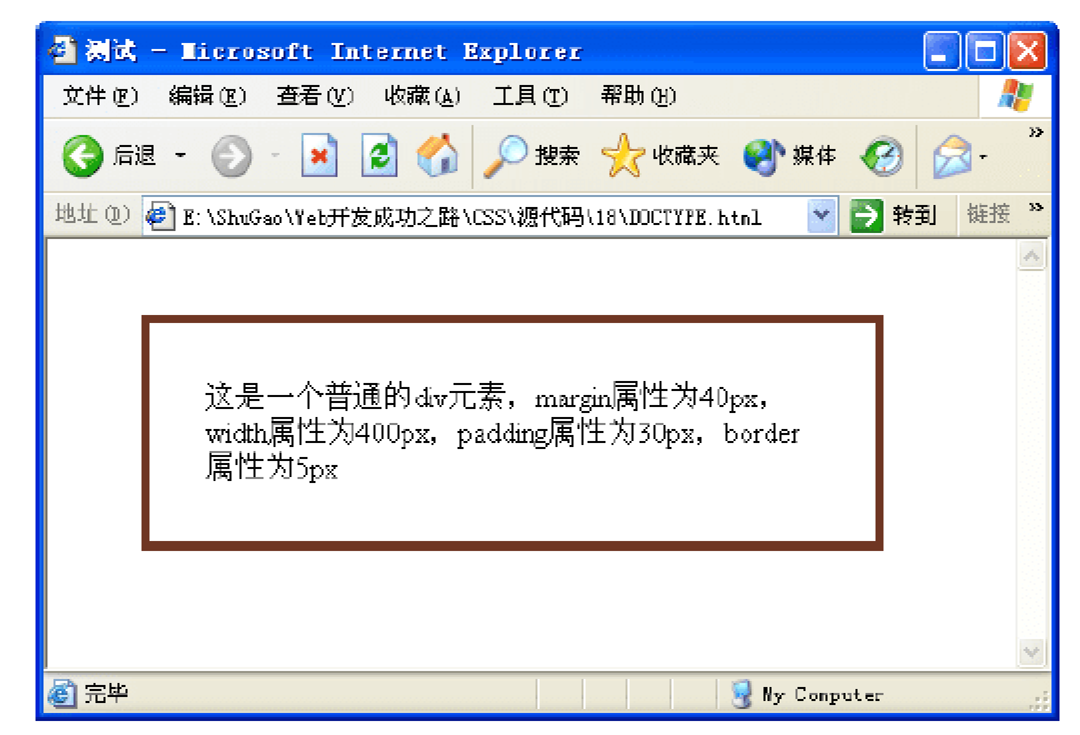
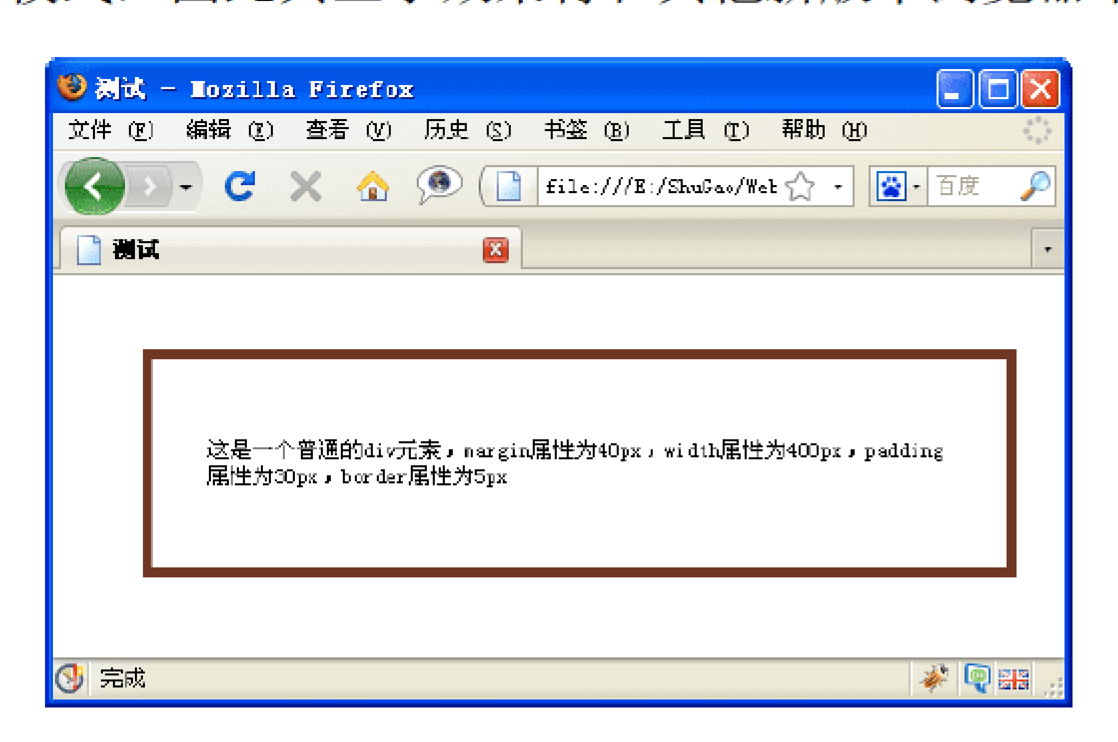


图 18-10　同一文档在 Firefox 与 IE 6 中的显示效果

这是因为在 DOCTYPE 的前面添加了 XML 声明(XHTML 文档的第一行)。如果将这个声明从文档中删除，那么 DOCTYPE 声明就成了该页面的第一个元素。此时，在 IE 中就会以标准模式呈现，如图 18-11 所示。而在 IE 7 中，即使添加了这个 XML 声明也不会切换到"怪异"模式，因此其显示效果将和其他新版本浏览器十分相似。



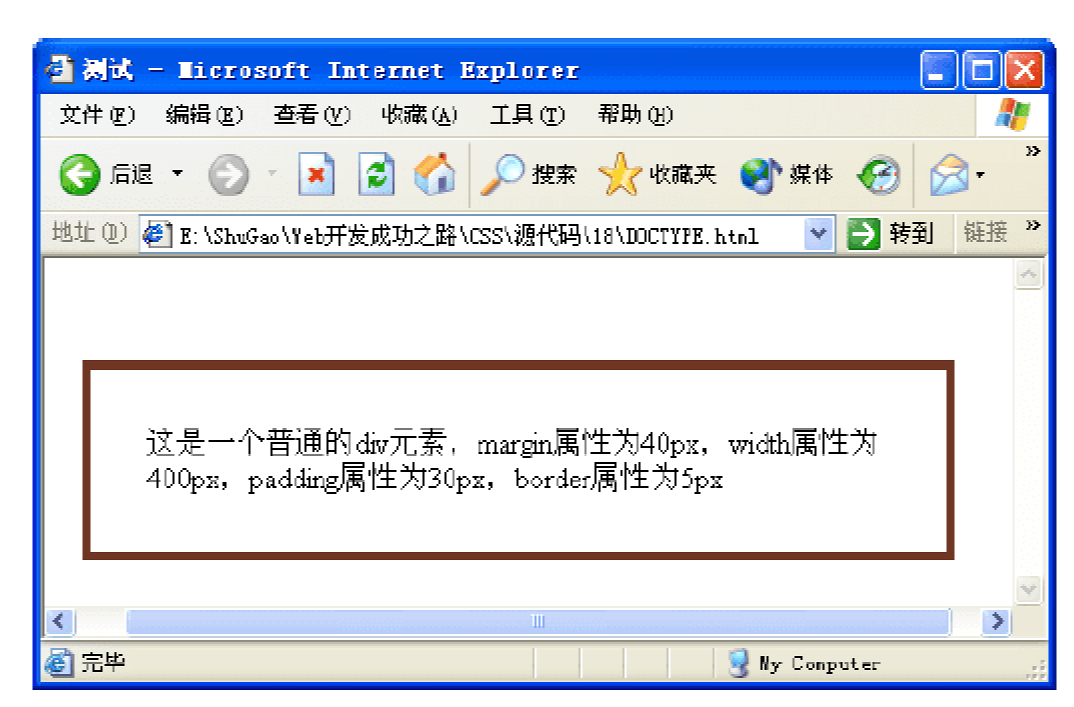


图 18-11 去除 XML 声明后，Firefox 与 IE 6 中的显示效果

18.4.2 基础知识——定义 DOCTYPE

如果你正在创建一个新网站，无论使用哪种 DTD，都建议采用标准模式。因为，在较新的浏览器中都支持 W3C 标准，而且无论是否支持 DOCTYPE 都会使用这些标准。

如果继续按照老版本浏览器的要求设计网页，并且依赖于其新版本的"怪异"模式，那么在那些不支持"怪异"模式的浏览器中就无法正常工作。最好的方法是按照新版本浏览器的标准模式来设计，而对于老版本浏览器可以使用本章介绍的兼容方法来修复。

使用如下的 DOCTYPE 声明可以强制浏览器支持 DOCTYPE 切换(包括 IE 6、Firefox 及 Opera 7)，使其转换到标准模式中。

- HTML 4.01 过渡性规范

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

- 支持框架的 HTML 4.01 规范

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

- XHTML 1.0 过渡性规范

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

- 支持框架的 XHTML 1.0 规范

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

● 严格的 XHTML 1.0 规范

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

● XHTML 1.1 规范

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```



即使在 DOCTYPE 语句上添加注释符号，也会使 IE 6 切换到“怪异”模式。

18.4.3 实例描述

如采用 XHTML 和 CSS 开发一个网站，然后在 IE 各版中对其进行测试，并且看起来一切正常。然后发现在基于 Mozilla 的浏览器中所呈现的页面布局与在 IE 中的效果有很大不同。

那么接下来怎么办呢？首先要检查的就是看看是否定义了 DOCTYPE 定义，因为它直接影响 W3C 标准是否被实现。

下面通过一个实例分析定义 DOCTYPE 前后页面效果的差别。

18.4.4 实例应用

【例 18-4】选择合适的 DOCTYPE

(1) 打开设计好的后台网站。在未添加 DOCTYPE 之前，先在浏览器中查看运行效果，如图 18-12 所示。

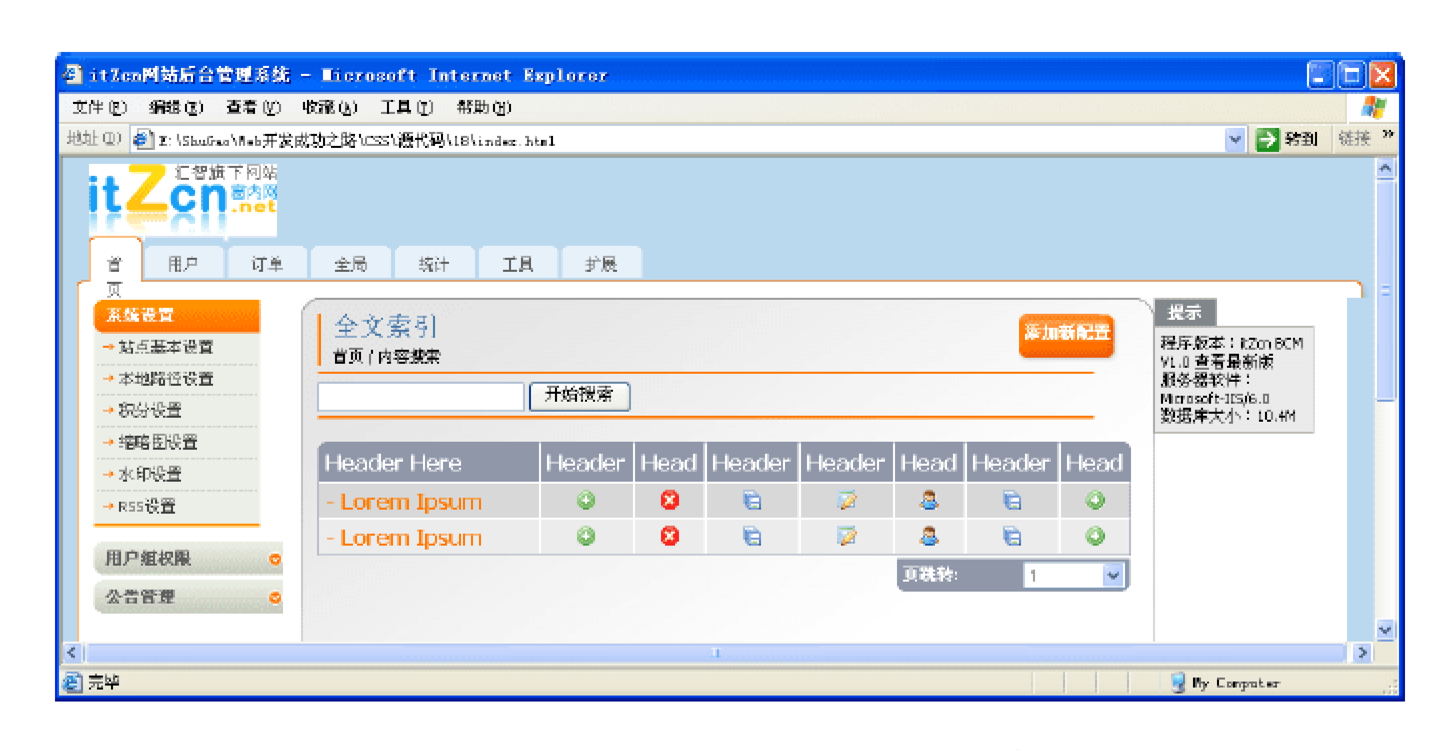


图 18-12 未使用 DOCTYPE 效果

(2) 从这张图中可以看到有很多显示错位或者未显示的地方。比较明显的有如下几处：

- “首页”链接应该在一行显示。
- “系统设置”、“用户组权限”和“公告管理”栏目右侧图片未显示，以及位置不对。
- 最右侧的背景没有平铺显示。
- 中部表格内的文字字号显示不正确。

(3) 打开页面的源代码，然后在<HTML>标签之前添加 DOCTYPE，并声明 XHTML 1.0 过渡性规范。代码如下所示：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

(4) 保存对 HTML 页面的修改，完成实例的制作。

18.4.5　运行结果

再次打开浏览器，查看使用 DOCTYPE 之后的运行效果，可以看到之前出现的不一致问题得到了修复，如图 18-13 所示。



图 18-13　使用 DOCTYPE 后的显示效果

18.4.6　实例分析



源码解析：

使用 CSS 设计网站时，如果发现在各浏览器中显示的效果差别很大，而且确认 CSS 没有错误时，添加 DOCTYPE 声明是一个不错的选择，它可以使页面遵循 W3C 标准，保持显示效果的一致性。

但是要注意<!DOCTYPE>标签没有结束标签。

18.5　CSS Bug

与许多编程语言相比，CSS 是一种相对容易学习的语言。它的语法简单明了，而且由于它的表现本质，开发人员不需要处理复杂的逻辑。但是，当在不同的浏览器上测试代码时，问题就出现了。浏览器 Bug 和不一致的显示方式是大多数 CSS 开发人员面临的主要难题。你的设计在一种浏览器上显示的很好，但是在另一种浏览器上布局可能会“支离破碎”。

“CSS 难于掌握”的误解并不来源于语言本身，而是由于为了让站点在所有主流浏览器上正常工作需要采取很多措施。关于 Bug 的信息很难找到，缺乏文档记录，因此 Bug 常常被误解。

许多人把 CSS 技巧看作魔术子弹，认为它们就像神秘的咒语一样，当应用于代码时，它们会神奇地修复破碎的布局。CSS 技巧的确是有力的工具，但是需要谨慎地应用它们，而且一般是作为最后的手段来使用。更为重要的技能是跟踪和识别 Bug。只有在知道一个 Bug 是怎么回事之后，才能想办法对付它。

18.5.1 校验 CSS Bug

我们都知道浏览器是有 Bug 的，当 CSS 开发人员在自己的代码中遇到问题时，他们往往马上就认为这是浏览器 Bug，并且应用适当的 CSS 技巧来解决。但是浏览器的 Bug 并没有一般人认为的那么常见。最常见的 CSS 问题并非来源于浏览器 Bug，而是来源于对于 CSS 规范的理解不完整。

许多开发人员是自学的 CSS，他们习惯按照自己的思维模型设计页面。当某些东西不符合他们的预期时，他们自然而然地把这归罪于浏览器，因此求助于某个 CSS 技巧来解决。为了避免这些问题，在处理 CSS Bug 时最好假设是做错了什么事。只有在确定这不是自己的错之后，才应该考虑问题是否来自浏览器 Bug。

其中，最简单的一些 CSS 问题是由代码中的输入错误和语法错误造成的。防止这种 Bug 的最好办法之一是通过 CSS 校验器运行代码。它会发现所有的输入错误，并且显示错误所在的行和对每个错误的简短描述。如图 18-14 所示为使用在线 CSS 校验器(http://jigsaw.w3.org/css-validator)的结果。



图 18-14　W3C CSS 校验器的结果

在校验自己的 XHTML 和 CSS 时，可能会看到页面充满了错误。这看上去会使人感觉非常沮丧，但是不必担心。这是因为大多数的错误是由一两个“真正”错误导致的结果。如果修复了第一个错误并且重新运行校验，可能会看到原来的很多错误消失了。这样修改几次后，代码很快就没有错误了。



校验器只是一个自动化工具，并不是完全可靠的。校验器报告的 Bug 数量正在增加，所以可能你认为某些东西是对的，但校验器却认为是错的，那么一定要检查最新的 CSS 规范。

18.5.2　捕捉 Bug 的基本步骤

捕捉 CSS 的 Bug 的第一步是检验 XHTML 和 CSS 是否存在输入和语法错误。某些显示错误是由于浏览器以“怪异”模式显示页面造成的。因此，最好检查是否正在使用适合自己的标记语言的 DOCTYPE，从而让页面以标准模型显示。关于这一点，可以使用 Firefox 浏览器很好地了解，方法是执行“工具”|“页面信息”命令，在弹出的窗口中查看渲染模式，如图 18-15 所示。



图 18-15　检查“怪异”模式或是标准模式

许多开发人员主要使用 IE 浏览器开发页面，所以每次修改页面之后，他们都会在 IE 中预览页面，看看效果是否正确。在页面基本完成之后，他们在各种浏览器中进行测试，并修复出现的任何 Bug。但是，这是一种危险的做法，而且可能会导致很多长期的问题。

这是因为 IE 5 有很多 Bug 和几个重要的 CSS 缺陷，包括它处理浮动的方式以及不正确的框模式实现。IE 6 的 Bug 稍微少一点，但是仍然有很多 Bug 和不一致的地方。由于使用 IE 作为主要的开发浏览器，很多开发人员错误地把 IE 的表现当作正确的表现，因此反而奇怪为什么标准浏览器“破坏了”他们精心设计的 CSS 布局。页面实际上是在 IE 中被“破坏了”，在标准浏览器中的显示方式是正确的。

更安全的方式是使用更符合标准的浏览器(例如 Firefox 或者 Safari)作为主要的开发浏览器。如果你的布局在这些浏览器中工作正常，那么你所做的很可能符合正确的方式。然后，可以在性能比较差的浏览器中测试页面，并且为发现的任何显示问题寻找解决方案。

1. 隔离问题

首先，我们来看看如何隔离问题。通过隔离问题和识别症状，可以查明是什么原因导致了这个问题并修复它。其中，隔离问题最有效的一种方法是在相关的元素上应用边框，看看它们的反应。

例如，如下是常见的边框样式：

```
#redBox{
    float:left;
```

```
    margin-right:5px;
    border:1px  solid red;
}
#blueBox{
    float:left;
    border:1px solid blue;
}
```

不过，我一般喜欢在代码中直接给元素添加边框。当然，也可以使用浏览器提供的工具来查看元素的轮廓。如果只通过添加边框就能修复问题，这往往说明存在空白边叠加的问题。

第二种方法，尝试修改几个属性，看看它们对 Bug 有什么影响，是以什么方式产生的影响。尝试使 Bug 的效果扩大化可能是有帮助的。例如，如果两个框之间的间隙在 IE 中比预期的大，那么就加大空白边，看看会发生什么情况。如果两个框之间的间隙在 IE 中加倍了，则可能是遇到了 IE 的双空白边浮动 Bug。

```
#redBox{
    float:left;
    margin-right:50px;
    border:1px  solid red;
}
#blueBox{
    float:left;
    border:1px solid blue;
}
```

另外，可以尝试一些常见的解决方案。例如，将 position 属性设置为 relative，将 display 属性设置为 inline(在浮动元素上)或者设置宽度和高度等尺寸，就可以修复很多 IE Bug。

2. 创建基本测试案例

如果问题比较棘手，就应该考虑创建一个基本的测试案例。

要创建基本的测试案例，首先应该复制出问题的文件。删除多余的 XHTML，直到只留下最基本的内容。然后注释所有样式表，从而查明是哪些样式表导致了这个问题。进入这些样式表，注释或者删除代码块。如果 Bug 突然消失了，那么就会知道是刚注释的最后一块代码导致了这个问题。继续注释掉代码块，直到只留下造成问题的代码。

然后，就可以开始详细研究这个 Bug。删除或者注释声明，看看会发生什么。这个修改对 Bug 有什么影响？修改属性值，看看问题是否消失了。使用一些常用的修复方法，看看它们是否有效。编辑 XHTML，看看是否有影响。使用不同的 XHTML 元素组合。有些浏览器有奇怪的空格 Bug，所以尝试从 XHTML 中删除空格。

3. 修复问题

知道问题的根源，对于实现正确的解决方案是非常有利的。因为给站点应用 CSS 样式有许多方式，最容易的解决方案是干脆回避这个问题。

如果空白边导致了问题，那么可以考虑用填充来代替。如果一种 XHTML 元素组合导致了问题，那么可以考虑换一种组合。

许多 CSS Bug 有描述性非常强的名称。所以在网络上搜索解决方案也是相当容易的。例如，

如果注意到 IE 将所有浮动元素上的空白边加倍了，那么搜索“IE 双倍边距 Bug”就会找到解决方案。

如果发现无法回避这个 Bug，那么可能不得不设法消除症状。这往往涉及寻找一个影响出问题的浏览器的过滤器，并且向这个浏览器提供单独的规则。

4. 寻求帮助

如果通过以上步骤还没有找到解决办法，那么就开始寻求帮助吧。即使是最有经验的开发人员，也可能遇到以前没有见过的问题。有时候和其他的开发人员探讨这些问题会帮助你解决它，即使他们可能无法马上给出具体的解决方案，可能也会提出一些新的值得一试的方法。

但是，如果你决定将问题发到论坛或者其他途径时，应该记住以下原则。

(1) 如果论坛提供了精华区，那么应该首先在精华区中搜索该问题，以免提出一个以前已经解决过的问题。

(2) 确保你的 CSS 和 XHTML 代码是有效的，否则你可能会得到类似“验证一下文档的有效性，看看是否有帮助”的回答。

(3) 将一个错误实例上传到帖子可以链接的位置。如果可以在一个复杂的页面布局之外重现这个问题，那是最好的。这样会使其他人更易于找到解决办法。

(4) 说明你已经尝试过的解决方法。这样，就可以节省那些看到你帖子的人的时间，避免陷入和你相同的死胡同，并说明在寻找帮助前你已经尝试着自己动手解决这个问题。

(5) 为你的帖子创建一个描述性的标题。标题为“在 IE 7 中的盒子模式重复显示了”的帖子要比“请求帮助！”更容易吸引人们的眼球，好的主题也能让精华区列表更有用，人们只看到标题就能大致了解其内容。

(6) 在描述内容时应该礼貌，内容应简明扼要。

(7) 在等待回答时要有耐心。如果一天之后还没收到任何回复，通常可以再发一次帖子。如果讨论版块特别大，帖子很可能没人注意到，这是一种提醒大家你的问题还没有收到任何回复的友好提示。

(8) 当你收到回复时，应尝试回复者的建议。如果他的建议不起作用，或者建议你尝试一些基本的方法，也不要生气。我经常看到很多帖子吸引大量的人参与，尝试帮助某人解决问题，直到找到解决方法为止。请给人们一个提供帮助的机会。

(9) 如果你找到了解决方法，或者决定修改设计以避免该问题，应该发一个帖子说明你是如何解决的，或者如何避免的。这样对帮助过你的人而言才显得礼貌，同时也对于那些遇到相同问题并从精华区中搜索信息的人更有帮助。当人们在精华区中搜索针对特定问题的解决方法时，如果找了几种可能的方法，但却不知道哪个是可行的，会有很大的挫折感。

18.6　CSS 建议

要想写出兼容性好，不容易出错的 CSS 样式规则，需要读者经常练习 CSS，在练习中逐渐掌握 CSS 规则，并形成习惯。下面给出了一些定义 CSS 规则时的建议及规范。

(1) 使用子选择器(descendant selectors)

CSS 初学者不知道使用子选择器是影响他们效率的原因之一。子选择器可以帮助你节约大

量的 class 定义。我们来看下面这段代码：

```
<div id=subnav>
<ul>
<li class=subnavitem> <a href=# class=subnavitem>Item 1</a></li>>
<li class=subnavitemselected> <a href=# class=subnavitemselected> Item 1</a>
</li>
<li class=subnavitem> <a href=# class=subnavitem> Item 1</a> </li>
</ul>
</div>
```

这段代码的 CSS 定义如下：

```
div#subnav ul { /* Some styling */ }
div#subnav ul li.subnavitem { /* Some styling */ }
div#subnav ul li.subnavitem a.subnavitem { /* Some styling */ }
div#subnav ul li.subnavitemselected { /* Some styling */ }
div#subnav ul li.subnavitemselected a.subnavitemselected { /* Some styling */ }
```

可以用下面的方法替代上面的代码：

```
<ul id=subnav>
<li> <a href=#> Item 1</a> </li>
<li class=sel> <a href=#> Item 1</a> </li>
<li> <a href=#> Item 1</a> </li>
</ul>
```

CSS 样式定义如下，这样用子选择器可以使代码和 CSS 更加简洁、更加容易阅读。

```
#subnav { /* Some styling */ }
#subnav li { /* Some styling */ }
#subnav a { /* Some styling */ }
#subnav .sel { /* Some styling */ }
#subnav .sel a { /* Some styling */ }
```

(2) 取消 class 和 id 前的元素限定

当你希望给一个元素定义 class 或者 id 时，可以省略前面的元素限定，因为 id 在一个页面中是唯一的，而 class 可以在页面中多次使用。仅仅限定某个元素毫无意义。例如：

```
div#content { /* declarations */ }
fieldset.details { /* declarations */ }
```

可以写成如下形式，省略一些字节。

```
#content { /* declarations */ }
.details { /* declarations */ }
```

(3) 默认值

通常 padding 的默认值为 0，background-color 的默认值是 transparent。但是在不同的浏览器中的默认值可能不同。如果怕引起冲突，可以在样式表一开始就先定义所有元素的 margin 和 padding 值都为 0，如下面代码所示：

```
* {
margin:0;
padding:0;
}
```

(4) 最近优先原则

如果对同一个元素的定义有多种，以最接近(最小一级)的定义为最优先，例如有这样一段代码：

```
Update: Lorem ipsum dolor set
```

在 CSS 文件中，已经定义了元素 p，又定义了一个 class update：

```
p {
margin:1em 0;
font-size:1em;
color:#333;
}
.update {
font-weight:bold;
color:#600;
}
```

这两个定义中，class=update 将被使用，因为 update 样式比 p 样式更接近 p 元素。

(5) 区分大小写

若在 XHTML 中使用 CSS，因 CSS 里定义的元素名称是区分大小写的。为了避免这种错误，建议所有的定义名称都采用小写。

class 和 id 的值在 HTML 和 XHTML 中也是区分大小写的，如果一定要大小写混合写，请仔细确认在 CSS 中的定义和 XHTML 中的标签是一致的。

(6) 明确定义单位，除非值为 0

忘记定义尺寸的单位是 CSS 新手普遍容易出现的问题。在 HTML 中可以只写 width=100，但是在 CSS 中，必须给一个准确的单位，例如：width:100px width:100em。只有两个例外情况可以不定义单位：行高和 0 值。除此以外，其他值都必须定义单位。注意，不要在数值和单位之间加空格。

(7) 不需要重复定义可继承的值

CSS 中子元素自动继承父元素的属性值(如颜色、字体等)，已经在父元素中定义过的。在子元素中可以直接继承，不需要重复定义。但是要注意，浏览器可能用一些默认值覆盖你的定义。

(8) 多重 class 定义

一个标签可以同时定义多个 class。例如：我们先定义两个样式，第一个样式背景为#666；第二个样式有 10 px 的边框。

```
.one{width:200px;background:#666;}
.two{border:10px solid #F00;}
```

在页面代码中，我们可以这样调用

```
<div class=one two></div>
```

这样最终的显示效果是这个 div 既有“#666”的背景，也有“10px”的边框。

(9) 使用 CSS 缩写。使用缩写可以帮助减少 CSS 文件的大小，更加容易阅读。

18.7 常见问题解答

18.7.1 CSS 样式表显示问题



CSS 样式表在不同电脑的浏览器上显示不正常。

网络课堂：http://bbs.itzcn.com/thread-11893-1-1.html

我的网站运用了大量的 CSS，在我的电脑上显示正常。可是放到同学的电脑上，用同样的浏览器看网页，结果显示就是错乱的，而在有的同学的电脑上是可以正常显示的。不知道这到底是什么原因。

【解决办法】一般是兼容性、编码格式、浏览器版本，还有浏览器设置方面的问题。根据这些看看你和你同学电脑中的两个浏览器有什么不同。

另外需要对不同的浏览器写不同的 CSS 样式(即 css hack)来兼容，使其效果一致，并在不同的主流浏览器上测试才行，如 IE 6 以上的版本、Firefox、opera、chrome、Safari 等。

18.7.2 CSS 固定 div 的兼容性问题



CSS 固定 div 的兼容性问题？

网络课堂：http://bbs.itzcn.com/thread-11894-1-1.html

我的 HTML 页面中代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>CSS 实现位置固定的 DIV</title>
<style type="text/css">
.aa{height:2000px;width:40px;background:#eee;}
.bb{width:50px;height:80px;background:red;float:right;position:fixed !impor
tant; top/**/:10px;position:absolute; z-index:100; right:20px;}
</style>
</head>
<body>
<div class="aa">
我是不动的
<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br
```

```
><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><b
r><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><
br><br><br><br><br><br><br><br><br><br><br><br><br><br>
</div>
<div class="bb">兼容 ff 和 IE</div>
</body>
</html>
```

在 IE 7 及 FireFox 中执行正常，为什么在 IE 6 中无法执行？

【解决办法】这个就是典型的浏览器兼容问题。

position:fixed !important 这里的 fixed 和 important 优先在 IE 6 中是无效的。所以，在 IE 6 中 bb 样式的 position 属性是后面定义的 position:absolute，只浮动而不固定位置。

18.7.3　简单菜单的兼容性



简单菜单的兼容性。

网络课堂：http://bbs.itzcn.com/thread-11895-1-1.html

我用 div+CSS 做的网页存在兼容性的问题，该如何解决呢？IE 6 中显示正常，而在 IE 7、IE 8 中都是乱的。

另外在调试网页时，IE 6、IE 7、IE 8 怎么才能装在一台电脑上进行调试？

【解决办法】现在大的网站都已经放弃 IE 6 了，也该是 IE 6 退出浏览器舞台的时候了，不过，国人还有很多在使用 IE 6。

其实 IE 8 的兼容性已经可以和 Firefox 相媲美，不过在实际应用过程中还应该针对 IE 8 做兼容工作。例如，如下的代码：

```
#tip {
background:blue;                /*Firefox 识别 */
background:blue\9;              /*ie8 识别 */
*background:green !important;   /*IE7 识别 */
*background:orange;             /*IE6 识别 */
}
```

按照这个在你的代码里一个一个调试。推荐一款叫 IEtester 的 IE 调试工具，非常不错，我一直在用，希望对你有帮助。

18.7.4　CSS 在 IE 和 FF 中的兼容性问题



CSS 在 IE 和 FF 中的兼容性问题。

网络课堂：http://bbs.itzcn.com/thread-11896-1-1.html

使用 CSS 设计了一个菜单，下面是菜单的主要样式和 HTML 代码：

```
     <style type="text/css">
      ul, li,a,h5{   padding:0; margin:0; }
```

```
        a{ cursor:pointer;}
        #nav li{  float:left; list-style-type:none; padding:5px;  }
        #nav li a {    display:block;  }
        #nav li ul{    *display:none; }
        #nav li a ul{  display:none;   }
        #nav li a:hover{ background:#fff;  }
        #nav li a:hover ul{ display:block; position:absolute; \left:10px;}
      </style>
   <h3>显示菜单</h3>
   <ul id="nav">
     <li>
        <a href="#">
        <h5>主菜单 1</h5>
        <ul>
          <li>子菜单项 11 </li>
          <li>子菜单项 12</li>
          <li>子菜单项 13</li>
          <li>子菜单项 14</li>
        </ul>
        </a>
     </li>
     <li>
        <a href="#">
        <h5>主菜单 2</h5>
        <ul>
          <li>子菜单项 21</li>
          <li>子菜单项 22</li>
          <li>子菜单项 23</li>
          <li>子菜单项 24</li>
        </ul>
        </a>
     </li>
  </ul>
```

有下面两个问题：

(1) 在上述代码中的 CSS 部分，如果把 #nav li ul{*display:none; }语句中的*号去掉，则可以在 Firefox 中隐藏 li 标记中的菜单。但是，当将鼠标移动到 a 标记上时，则#nav li a:hover 伪类将不显示。

(2) 把 a:hover 里的 background:#fff;去掉，或改成其他属性，则不正常，为什么？

【解决办法】先不说其他的，一看代码就明白了。标题样式是不能用在列表中的，如果需要使用同标题相同的样式，那就直接用 CSS 来设置样式。

而你却把标题这么重要的标签放在
和
之间了。

```
<li>
  <a href="#">
  <h5>aaaaaa</h5>
  <ul>
  <li>aaaaaa </li>
```

```
  <li>aaaaaaf </li>
  <li>a11111 </li>
  <li>a11111 </li>
  </ul>
  </a>
</li>
```

这样做是不允许的。当然，也不是说你这样写就一定不行。但由于浏览器的不同带来的样式和效果也不相同。

这不是严格不严格的问题，而是兼容不兼容的问题。你的网页要想让更多的人看就一定要尽可能地减少这种代码上的错误。如果代码正确率高，给不同浏览器带来的差别就会小很多。这样，就算是有了小问题，自己试试或到这里问问，也就能很快解决了。

18.8　习　　题

1. 填空题

(1) 使用 CSS 条件注释中的__________表达式可以判断 IE 浏览器版本是小于或等于某个版本。

(2) 只有普通文档流中区块的垂直空白边才会发生叠加，对于__________、浮动块或__________之间的空白边是不会叠加的。

(3) 使用__________来解决闭合浮动元素时，需要设置为 hidden 或者 auto。如果在 IE6 下，还需要指定高度或者__________。

(4) DOCTYPE 可声明三种 DTD 类型，分别表示严格版本、__________以及基于框架的 HTML 文档。

2. 选择题

(1) 使用__________可以区分 IE 浏览器版本的语法。

A. 空白边叠加　　　　B. 闭合浮动元素

C. DOCTYPE　　　　D. 条件注释

(2) 如果当前浏览器版本大于 6，则连接 other.css，应该选择__________。

A.

```
<!--[if gte IE 6]>
  <LINK href="other.css" type="text/css" rel="stylesheet">
<![endif]-->
```

B.

```
<!--[if IE 6]>
  <LINK href=" other.css" type="text/css" rel="stylesheet">
<![endif]-->
```

C.

```
<!--[if gt IE6]>
  <LINK href=" other.css" type="text/css" rel="stylesheet">
<![endif]-->
```

D.

```
<!--[if lt IE 6]>
  <LINK href=" other.css" type="text/css" rel="stylesheet">
<![endif]-->
```

(3) 当两个垂直空白边相遇时，它们将形成一个空白边。这个空白边的高度等于两个发生叠加的空白边中的__________。

A. 较小者　　B. 较大者　　C. 平均值　　D. 之和

(4) 浮动元素之所以未闭合是由__________属性引起的。

A. float　　B. overflow　　C. position　　D. padding

(5) 要实现闭合浮动元素，下面__________方式不可选。

A. 使用<div style="clear:both;"></div>　　B. 使用 after 伪类

C. 浮动外部元素　　D. 使用 width:100%

(6) 支持框架的 XHTML 1.0 规范使用 DOCTYPE 应该遵循__________DTD。

A. http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd

B. http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd

C. http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd

D. http://www.w3.org/TR/html4/frameset.dtd

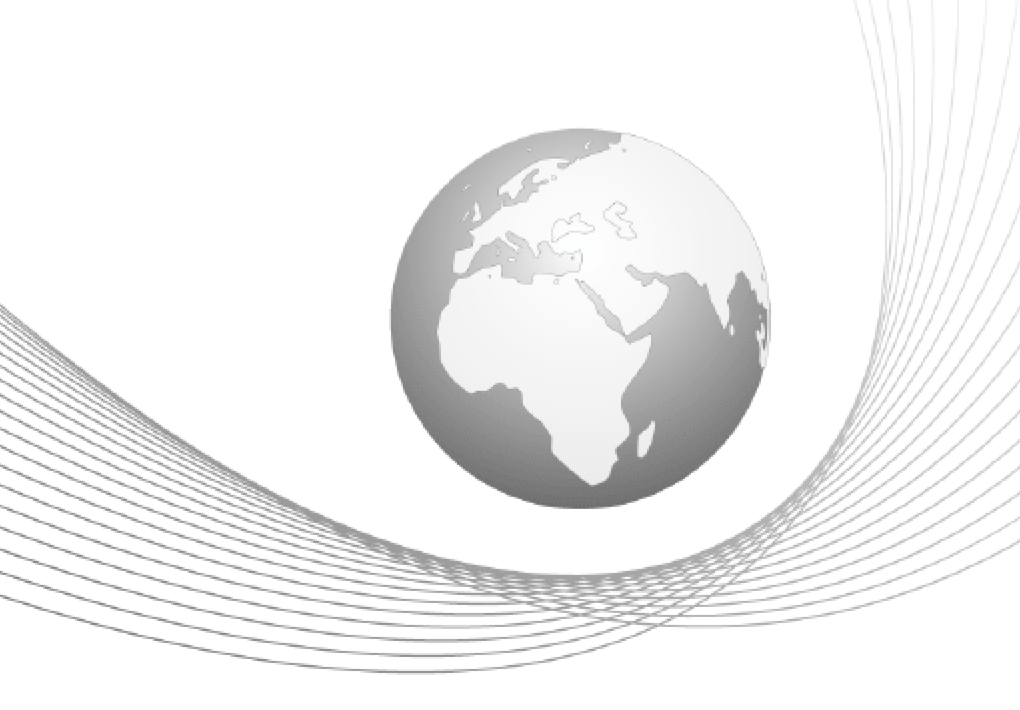
3. 上机练习

上机练习：解决页面显示不正常的问题

现在有一个 HTML 页面，在 IE 8 中的运行效果如图 18-16 所示。从图中可以看到页面显示不正常，这就要求读者应用本章所学到的知识，在不改变 HTML 的布局的情况下进行修复。



图 18-16　运行效果



第 19 章　itzcn 书店设计

内容摘要：

窗内网(www.itzcn.com)是一个免费的在线视频教育平台，它结合了软件开发的实际需求，发布了 Flex、C#、Java、ASP.NET、PHP 和 JSP 等系列视频教程。与此同时，线下的图书编辑业务也在大量推广。很多同学因为看过在线视频以后找不到对应的教材进行同步学习，所以他们在网站上提出了建一个在线书店的建议，甚至有热心的网友帮我们设计好了页面的风格。

下面我们就把网友提供的页面设计图使用 div+CSS 技术做成一个静态的页面，也同时练习一下我们学习的 CSS 技术。

学习目标：

- 掌握页面布局的设计方法
- 掌握使用 CSS 控制 div 布局的方法
- 掌握元素定位的方法
- 掌握使用 CSS 对页面对象设置样式的方法

19.1　案例效果图分析

HTML 页面其实是由一个个文字、图片、动画等基本元素组成的。将它们放置在页面标签中，再使用 CSS 样式设置以后才能形成美轮美奂的页面。

从拿到一张页面设计图开始，我们就定下了一个目标：要使用 CSS 样式操纵页面元素组成一个漂亮的艺术品。

我们首先要对这张设计图进行分解，一块块地划分，直至分成最小的再不可分割的元素为止。就像一个拿着刻刀的孩子，一刀刀地把这张图切割成我们想要的结果。

下面我们就来实施这个计划。页面设计的最终效果如图 19-1 所示。



图 19-1　页面效果图

19.1.1　划分页面主体框架

通过仔细分析图 19-1 可以看出，页面中所有的内容都居中显示在中间 900 像素宽的内容区域，所以需要创建一个总的内容容器用于盛放所有的元素。

页面中的内容区域可以根据各区域间明显的界线分为上、中、下三个大的区域，其中上面

的区域包含的内容如图 19-2 所示。



图 19-2　页面顶部

下面的区域包含的内容如图 19-3 所示。



图 19-3　页面底部

当然，剩余的就是中间区域的内容了。

总体来说整个页面的主体框架如图 19-4 所示。

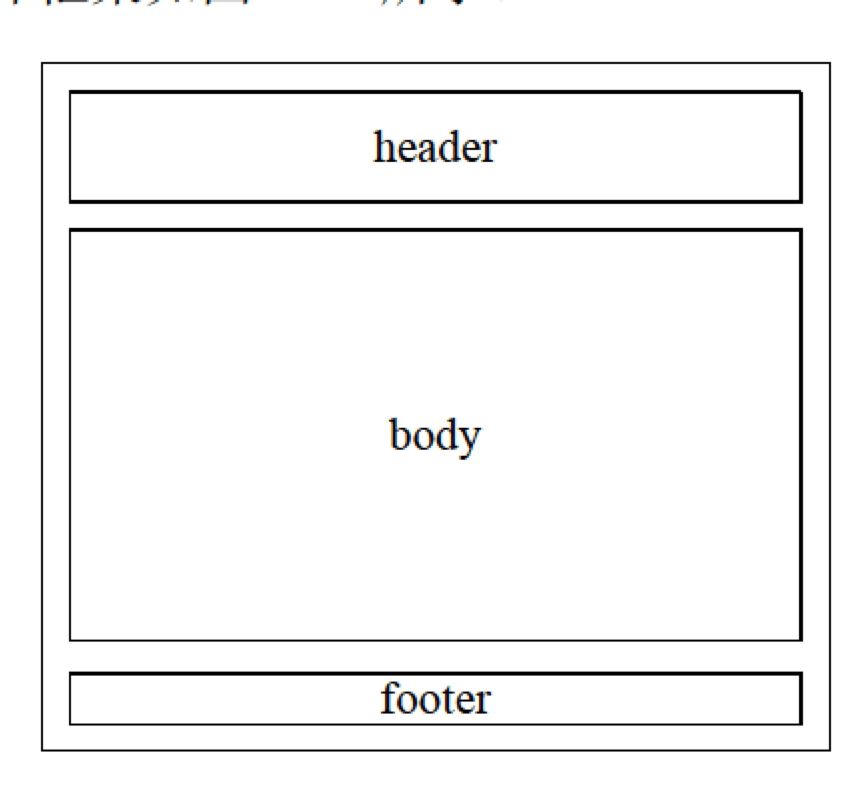


图 19-4　页面主体框架

19.1.2　分析顶部模块

页面顶部区域结构非常简单，主要分为上部的形象区域和下面的菜单条两部分。

当然，上面的形象区域与页面浑然一体，不可能会经常变化，所以可以整体作为一张图片进行处理，简单方便。不过作为站点 Logo 的“itzcn 书店”字样需要单独处理，所以要求可以单独操作。而下面的菜单条可以设计为一个横向排列的列表。所以，整个顶部模块的页面结构如图 19-5 所示。

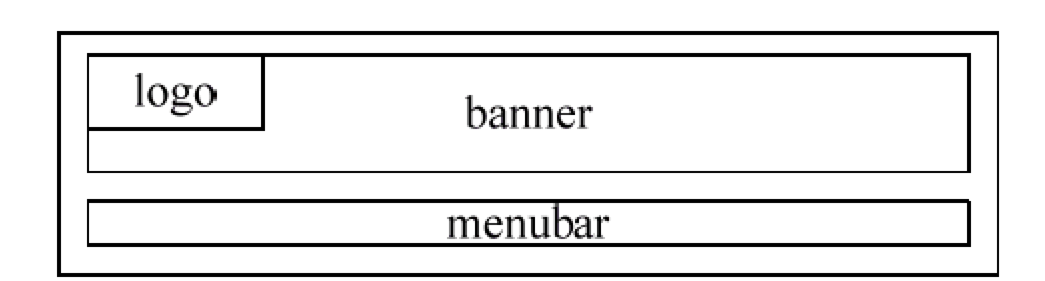


图 19-5　页面顶部结构

19.1.3　分析底部模块

底部模块更加简单，主要有靠近左侧的 Logo 展示区和靠近右侧的导航链接区。然后就是底下一条呼应顶部主题的修饰线条了。

所以，整个底部的结构如图 19-6 所示。

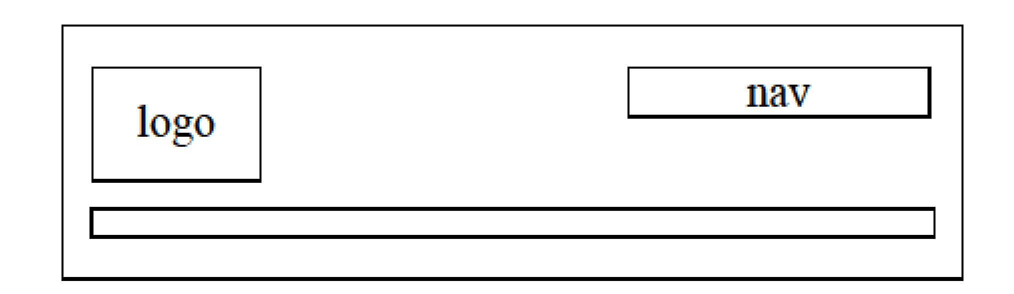


图 19-6　页面底部结构

19.1.4　分析页面主体部分结构

页面中间内容区域的结构比较清晰，但是由于内容较多，所以相对来说也比较复杂。但是从直观感觉上可以先分为左、右两大部分，如图 19-7 所示。

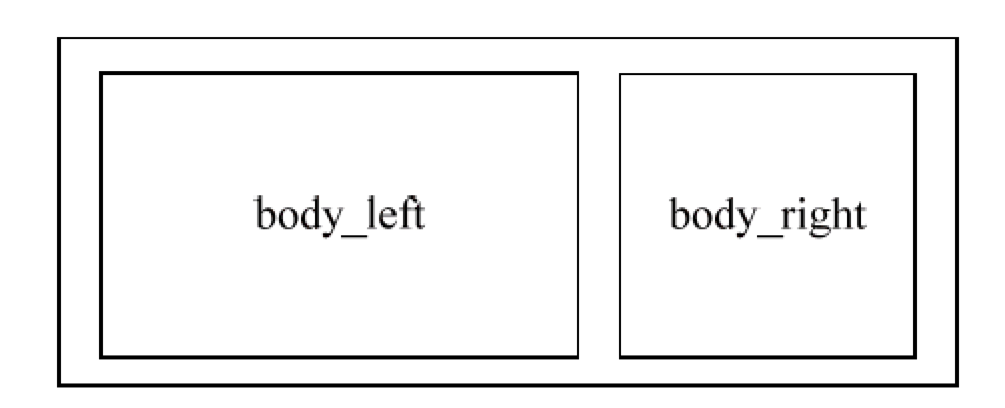


图 19-7　页面内容区主体框架

不过，由于页面主体部分内容背景比较统一，可以纵向重复，所以为了节省代码，可以将页面主体部分作为一个整体设置背景。这样不仅便于管理，也节省了代码量。

1. 页面主体部分左侧结构设计

页面主体部分左侧的内容结构比较简单，只需要“好书推荐”和“新书上架”两个模块。在“好书推荐”模块中，两条记录纵向显示，所以不用刻意进行布局；而“新书上架”模块中有三项数据，横向排列，这里可以使用一个容器来存放它们。

所以页面主体部分左侧结构可以设计成如图 19-8 所示的页面布局。



这里为了节省篇幅，就不再深入地对每一个小模块进行分解了，详细的分析会在页面具体实现的时候进行讲解。

2. 页面主体部分右侧结构设计

相对于主体部分的左侧结构来说，右侧模块有些复杂。

首先它有 5 个主要的栏目：“购物车”、“关于书店”、“优惠图书”、“图书分类”和“友情链接”，而且每个栏目的内容也不是很多，所以布局设计得比较紧凑，结构也比较合理。

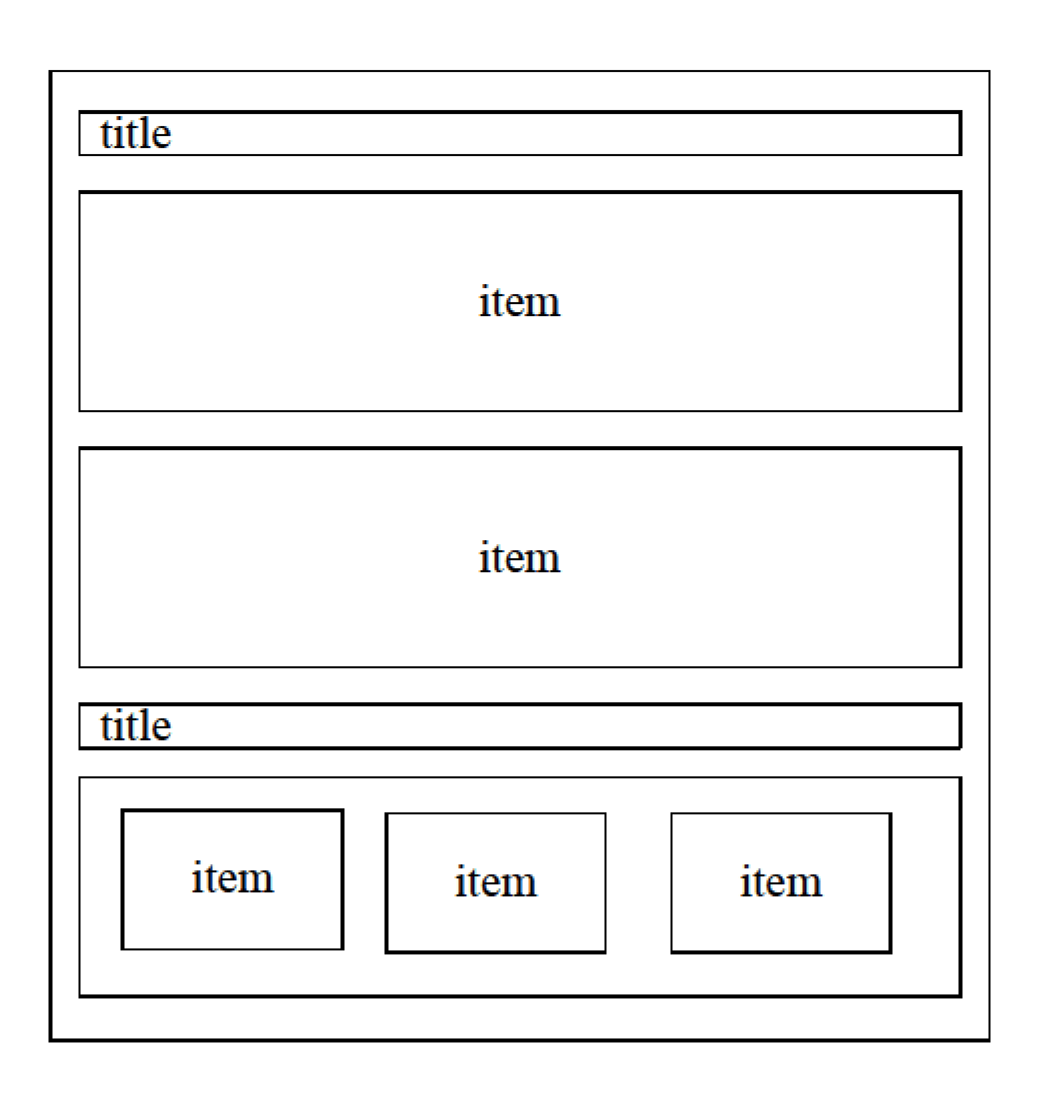


图 19-8　左侧布局设计

总体来说，右侧共分为上、中、下三大块，其中下面又分为左、右两个部分，然后右边又分为上、下两个模块，比较纠结。

不过这些模块并不复杂，而且界线比较清晰，可以直接使用五个模块进行拼凑组合到一起来完成这块布局。

所以，页面主体部分右侧的模块结构可以设计成如图 19-9 所示的样子。

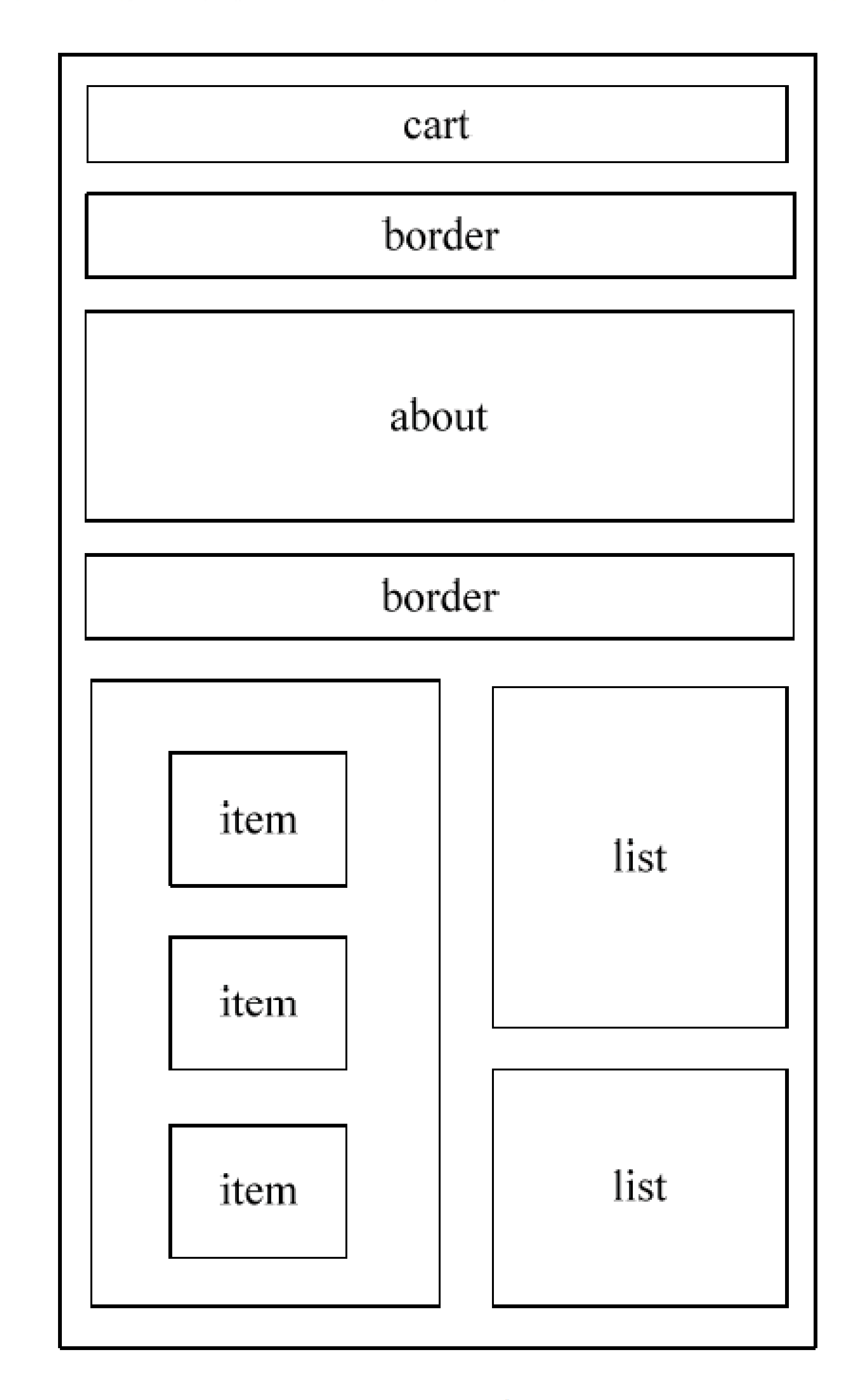


图 19-9　右侧布局设计



或许这种方式并不是最佳方案，它没有多级嵌套容易控制，但这样会在很大程度上简化页面结构，所以我们采用这种方式来实现。

19.2 站点目录结构

要想很好地理解下面将要讲的程序，我们必须先了解站点的目录结构。

本实例使用的开发工具是 Dreamweaver。首先在 Dreamweaver 中的文件管理窗口中新建一个站点，名为 ItzcnBookStore。该站点下面有三个目录，用于分别存放页面中的资源文件。

下面看一下 Dreamweaver 中的文件管理窗口中的目录结构，如图 19-10 所示。

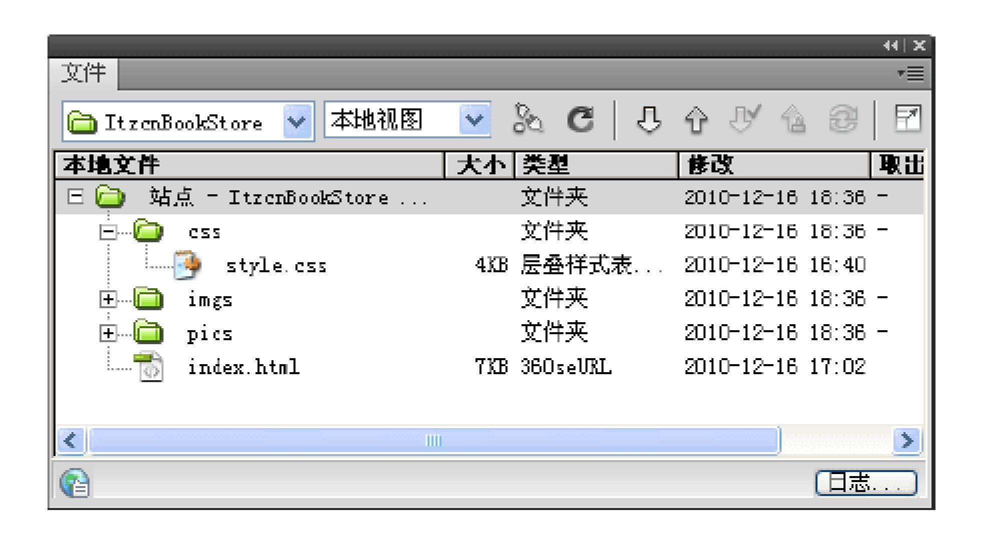


图 19-10 目录结构

文件管理窗口中的三个子目录的作用说明分别如下。

- css：用于存放 CSS 样式表文件。
- imgs：用于存放控制页面风格要用到的样式图片。
- pics：用于存放网页中的内容图片。

最后，站点的 HTML 页面文件统一放置在站点根目录下。

19.3 页面框架布局设计

前面说过，我们的页面框架总体分为上、中、下三个大的模块，在中间的内容区域模块中又主要分为左右两个子模块。所以，页面主体框架可以设计成如图 19-11 所示的结构。

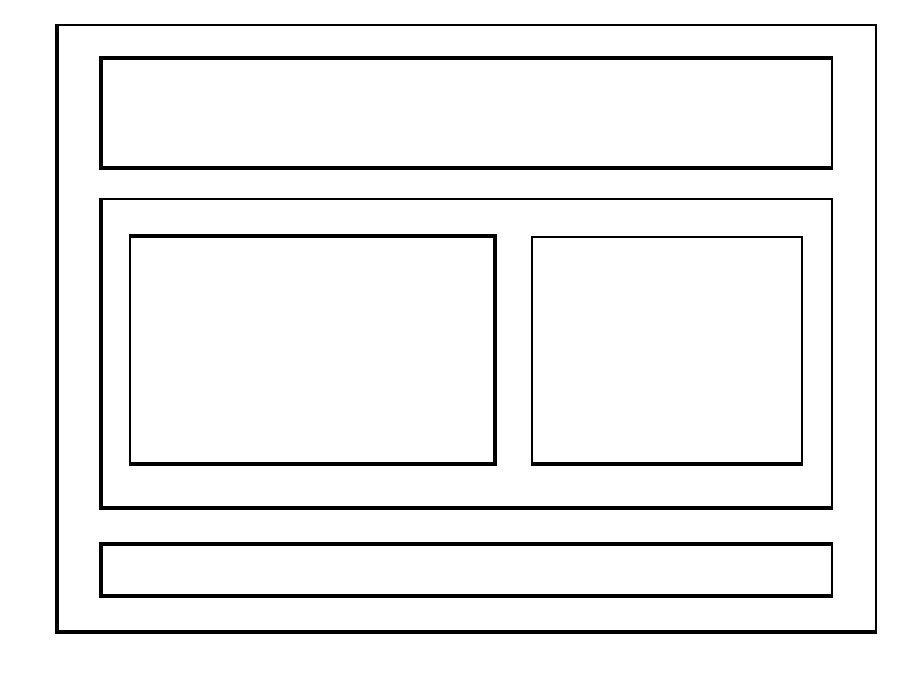


图 19-11 页面主体框架结构

根据图中的结构设计，可以在页面中添加如下的 HTML 代码：

```
<div id="container">
  <div id="header"> </div>
  <div id="body" class="clear">
    <div id="body_left"> </div>
    <div id="body right"> </div>
  </div>
  <div id="footer"> </div>
</div>
```

在页面中，container 是一个主容器，里面包含 header、body 和 footer 三个子容器，其中在 body 容器中又包含 body_left 和 body_right 两个容器。



因为 body 容器中的两个子容器需要横向显示，并将会为它们设置浮动效果。为了处理浮动后的页面布局混乱的问题，我们为 body 容器添加了一个 clear 样式，用于统一处理页面中包含浮动元素的容器样式。

接下来需要为页面主框架中的元素设置相应的样式。

首先在 css 目录中创建一个样式表文件，可以根据各自的需求命名，这里命名为 style.css。然后在页面中的 head 部分使用 link 元素引入该样式表文件，代码如下：

```
<link rel="stylesheet" type="text/css" href="css/style.css">
```

然后在样式表中添加相应的 CSS 样式代码。

下面代码是对整个页面进行的一些设置：

```
html, body, div, ul, img {
    margin:0;
    padding:0;
}
img {
    border:0;
}
body {
    background-color:#6F605A;
    font-family:"宋体", Arial;
    font-size:12px;
}
.clear:after {
    content:".";
    clear:both;
    display: block;
    height:0;
    visibility:hidden;
}
```

这里设置了页面中的 html 元素、body 元素、div 元素、ul 元素、img 元素的边距和填充默认值都为 0；然后设置了 img 元素的边框为 0，避免使用超链接后出现一个难看的蓝框；接着

设置页面的背景色、字体、字号等属性；最后定义了一个 clear 类样式，设置最后一个子元素的内容为一个圆点，并清除两侧的元素浮动属性，显示为一个块元素，高度为 0，而且将其设置为隐藏。

下面我们再为页面框架添加 CSS 样式。

首先页面主容器元素的样式定义如下：

```
#container {
    width:900px;
    margin:auto;
    background-color:#FFFFFF;
}
```

这里设置页面内容区宽度为 900 像素，边距为自动，背景色为白色。

然后设置页面头部的样式，代码如下：

```
#header {
    background-image:url(../imgs/header.jpg);
    height:181px;
}
```

这里定义了页面头部的背景和高度。因为 div 宽度默认是 100%，所以不用控制它的宽度属性。

接下来需要设置页面主体部分的样式，代码如下：

```
#body {
    background-image:url(../imgs/center_bg.gif);
}
```

页面主体部分内容不固定，所以只需要设置一个背景图片就行了。

下面再来看一下页面主体部分左侧模块的样式代码：

```
#body_left {
    padding:20px;
    width:471px;
    float:left;
}
```

设置宽度为 471 像素、填充值为 20 像素，并且向左侧浮动。

然后来看一下页面主体部分右侧模块的样式代码：

```
#body_right {
    padding:20px;
    width:349px;
    float:right;
}
```

同样设置了宽度、填充值和浮动属性，不过这里是向右浮动的，也就是说整体向右靠，使右侧与页面右边对齐。

最后来看一下页面脚注部分的代码：

```
#footer {
    height:101px;
    border-top:dashed 1px #bbb;
}
```

这里只设置了区域高度为 101 像素，并且上边框为一条 1 像素宽的灰色虚线。

设置完以后，页面效果如图 19-12 所示。

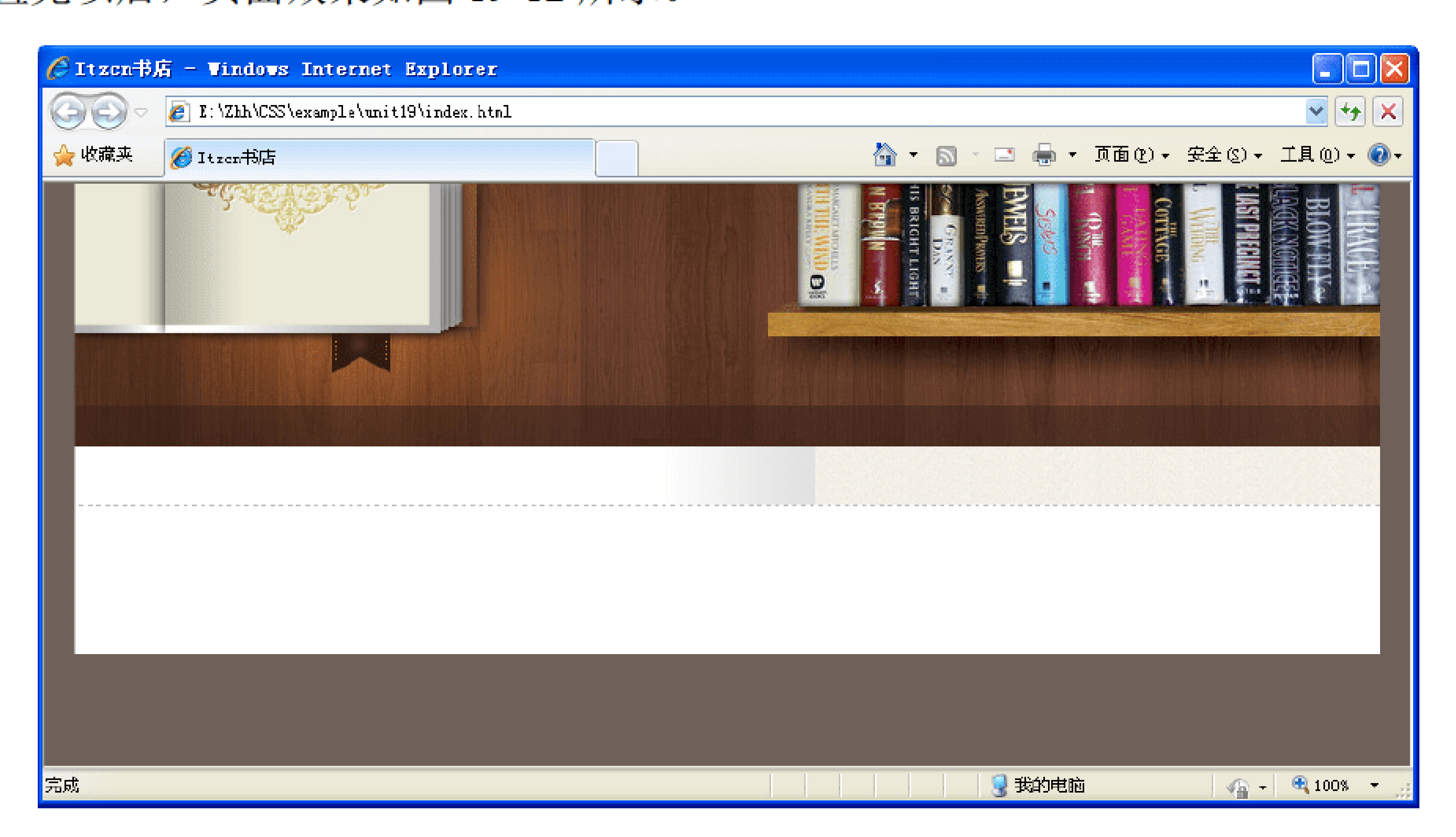


图 19-12　成型的页面主体框架

19.4　头部 banner 和菜单条的布局设计

我们已经把主体框架搭起来了，然后再对每个模块添加内容就相对容易多了。

经过 19.3 节的分析研究，顶部就差一个 Logo 和菜单条了，因为顶部背景图片已经给出了菜单条的位置，所以这里需要定义两个容器来分割页面头部的空间。

第一个 banner 的空间占据顶部除菜单条以外的内容，包括 Logo；第二个放置一个列表元素，列出页面顶部的菜单条，所以顶部内容在页面中的 HTML 代码如下：

```
<div id="header">
  <div id="banner">
    <h1 class="Logo">Itzcn 书店</h1>
  </div>
  <!-- ============= 菜单条(开始) ============= -->
  <div id="menubar">
    <ul>
      <li class="active_menu"><a href="#">首页</a></li>
      <li><a href="#">关于书店</a></li>
      <li><a href="#">新书列表</a></li>
      <li><a href="#">图书专辑</a></li>
      <li><a href="#">我的账号</a></li>
      <li><a href="#">用户注册</a></li>
```

```
      <li><a href="#">联系我们</a></li>
      <li><a href="#">人才招聘</a></li>
    </ul>
  </div>
  <!-- ============= 菜单条(结束) ============= -->
</div>
```

上面代码在 header 元素中定义了一个 banner 子容器和一个 menubar 子容器。在 banner 中，添加了一个一级标题，作为显示的 Logo；在 menubar 中添加了一个列表元素，每一个列表项中添加一个超链接。

下面我们来看它们的样式声明。

首先是 banner 的样式，代码如下：

```
#banner {
    height:152px;
}
#banner .Logo {
    color:#604329;
    height:82px;
    display:block;
    padding:55px 0 0 95px;
    margin:0;
    font-size:30px;
    font-weight:800;
    font-family:"黑体", "Arial Black";
}
```

这里首先设置了 banner 元素的高度为 152 像素(当然剩下的就是菜单条的高度了)；然后设置了作为 Logo 的页面元素 h1 的样式，其中有颜色、高度、显示模式、填充、边距、字体大小、字体重量、字体等。

下面是菜单条 menubar 的样式，代码如下：

```
#menubar {
    height:24px;
    padding:5px 0 0;
}
#menubar ul {
    margin:0 0 0 30px;
}
#menubar li {
    display:inline;
    padding:0 10px;
}
#menubar li a {
    font-size:14px;
    text-decoration:none;
    color:#fff;
}
.active_menu a {
```

```
    color:#DDA83A !important;
}
```

上面代码首先设置菜单条的高度和左填充；然后设置菜单列表的右边距；再设置菜单项的显示方式为 inline(在一行中显示)，并设置该项的左右填充值为 10 像素；接着设置菜单项里的超链接的字体、文本装饰、字体颜色等样式；最后添加一个活动菜单项的样式，修改一下菜单链接文本的颜色。不过这里要注意 CSS 选择器“.active_menu a”没有选择器“#menubar li a”的优先级高，所以需要设置一个“!important”关键字来增加该样式的优先级。

下面运行看看效果，如图 19-13 所示。



图 19-13　页面顶部设计的效果

19.5　底部页面脚注的布局设计

页面底部的内容非常简单，可以简单地分为三块，第一块是两个图片，第二块是一个导航链接，第三块是一张图片。所以可以这样设计页面底部的页面代码：

```
<div id="footer">
  <div class="Logo"> <img src="imgs/Logo.png" /> <img src="imgs/itzcn.gif" />
</div>
  <div class="nav">
    <ul>
      <li><a href="#">首页</a></li>
      <li><a href="#">关于我们</a></li>
      <li><a href="#">服务</a></li>
      <li><a href="#">保密协议</a></li>
      <li><a href="#">联系我们</a></li>
    </ul>
  </div>
  <img src="imgs/footer_bg.gif" />
</div>
```

页面结构就不用解释了，直接来分析下面的样式代码：

```
#footer .Logo {
    height:70px;
    width:100px;
    padding:10px;
    float:left;
}
#footer .nav {
    float:right;
    padding:10px;
}
#footer .nav li {
    display:inline;
    margin-right:10px;
}
#footer .nav li a {
    color:#aaa;
    font-size:13px;
    text-decoration:none;
}
```

这里首先设置 Logo 区的大小、填充和浮动属性；然后设置导航的浮动属性和填充属性；再设置导航项目的显示属性和右边距属性；最后还要定义页面导航链接的颜色、字号、文本修饰等属性。

设置完成以后查看运行效果，如图 19-14 所示。



图 19-14　页面脚注设计的效果

19.6　左侧主内容区域的详细设计

窗口左侧主要显示两个页面栏目：“好书推荐”和“新书上架”。这两个栏目的结构互不影响，所以我们可以分开设计。

19.6.1　“好书推荐”栏目样式设计

该栏目包括一个标题和两个图书信息。

标题是由一个图标和一个标题文本组成的，而且从效果图中可以看出，页面中 7 个栏目的标题样式完全相同，所以可以统一为它们共同设置一个样式(以后如果没有必要的话就不再对它们进行讲解)。

首先选择使用 h2 元素显示这个标题，比如当前栏目中的标题代码：

```
<h2><img src="imgs/bullet1.gif" />好书推荐</h2>
```

可以为 h2 这个元素定义一个 CSS 样式，代码如下：

```
h2 {
    color:#734633;
    font-family:"黑体", "Arial Black";
    font-size:21px;
    font-weight:700;
    margin:15px 0;
}
h2 img {
    margin-right:5px;
}
```

这样先设置栏目标题的颜色、字体、字号等属性，然后再设置标题内右边距为 5 像素。

两个推荐图书的项目样式完全一样，所以可以为它们设置同样的页面结构。

首先每个推荐图书的项目分为左、右两块，可以设置横排的两个元素。其中左侧的模块只是一个图像文件，右侧的模块是一个灰线边框的圆角矩形括起的内容，所以可以将圆角矩形的上边和下边作为单独的元素显示，中间设置同样宽度的背景图片即可。同样，中间的内容也可以分为三部分：图书名称、说明和详细说明链接，所以可以再分为上、中、下三个元素。整体结构如图 19-15 所示。

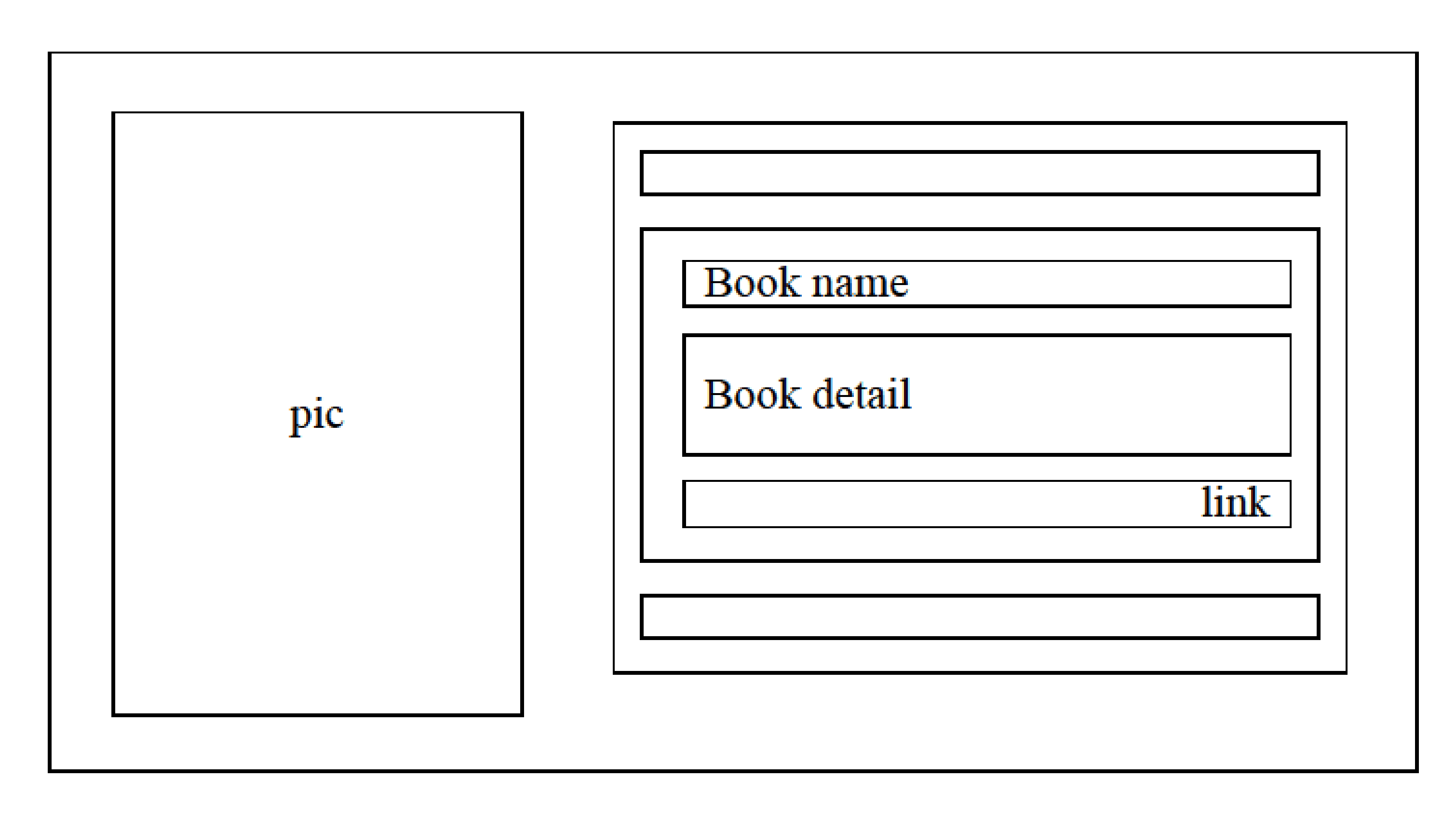


图 19-15　“好书推荐”栏目的样式设计

所以，整个“好书推荐”栏目的 HTML 代码如下：

```
<h2><img src="imgs/bullet1.gif" />好书推荐</h2>
<div class="prod clear">
  <img src="pics/G8100758.jpg" class="pic" />
  <div class="explain_box">
    <div class="top"></div>
    <div class="content">
      <div class="title"><a href="#">电影风暴：Maya 游戏场景大曝光</a></div>
      <div class="details">本书介绍利用 Maya 软件开发网络游戏的实现技术。内容包括 Maya
的基础知识，建模技术，材质、灯光的应用；游戏场景、游戏角色、游戏特效设计。</div>
      <div class="more"><a href="#">- 详细说明 -</a></div>
    </div>
    <div class="bottom"></div>
  </div>
</div>
<div class="prod clear">
  <img src="pics/G7024623.jpg" class="pic" />
  <div class="explain_box">
    <div class="top"></div>
    <div class="content">
      <div class="title"><a href="#">SQL Server 2008 完全学习手册</a></div>
      <div class="details">本书从数据库的基础知识入手，全面介绍 SQL Server 2008 数据
库应用知识。全书内容包括数据查询和管理，存储过程，触发器，安全机制，集成服务，分析服务和报
表服务等内容。</div>
      <div class="more"><a href="#">- 详细说明 -</a></div>
    </div>
    <div class="bottom"></div>
  </div>
</div>
```

这段 HTML 代码的样式声明代码如下：

```
.prod {
    padding:10px;
    margin-bottom:15px;
    border-bottom:dashed 1px #bbb;
}
.prod .pic {
    float:left;
    width:98px;
    height:150px;
}
.prod .explain box {
    width:295px;
    float:right;
    margin-right:25px;
}
.prod .explain_box .top {
    height:9px;
    background-image:url(../imgs/box_top.gif);
}
.prod .explain_box .content {
```

```
    height:100px;
    padding:10px;
    background-image:url(../imgs/box center.gif);
}
.prod .explain box .bottom {
    height:9px;
    background-image:url(../imgs/box_bottom.gif);
}
.prod .explain_box .content a {
    color:#F9981D;
    text-decoration:none;
}
.prod .explain_box .content .title {
    font-size:15px;
    height:20px;
}
.prod .explain_box .content .details {
    height:65px;
    line-height:18px;
}
.prod .explain box .content .more {
    text-align:right;
    height:15px;
}
```

上面样式表中，首先设置模块的内填充属性、底部边距属性和底部边框属性；然后设置左侧图片的宽、高和浮动属性；再设置右侧说明的右边距、高度和浮动属性。对于右侧模块的顶部设置元素的高度和背景图片实现圆角边框的上边框样式；中间的元素设置其高度、填充和背景属性；底部的元素设置其高度和背景实现圆角边框的下边框样式。然后设置说明框中的链接颜色和文本修饰；再设置说明内容的标题(即图书名称)元素的字体和高度；接着设置说明内容信息的高度和行高样式。最后设置"查看更多"链接所在元素的文本对齐方式和元素高度属性。

完成这些样式的设置以后，浏览器运行的页面效果如图 19-16 所示。



图 19-16　"好书推荐"栏目的运行效果

19.6.2 “新书上架”栏目样式设计

“新书上架”栏目模块比较简单，没有太复杂的页面结构，只需要设置一个父容器，父容器中放置三个横排的项目即可。横排的项目每个都可以分为上、下两部分，上面是书的名称，下面是图片，不过这里的图片需要在右上角添加一个标识。所以，页面结构可以设计成如图 19-17 所示的样式。

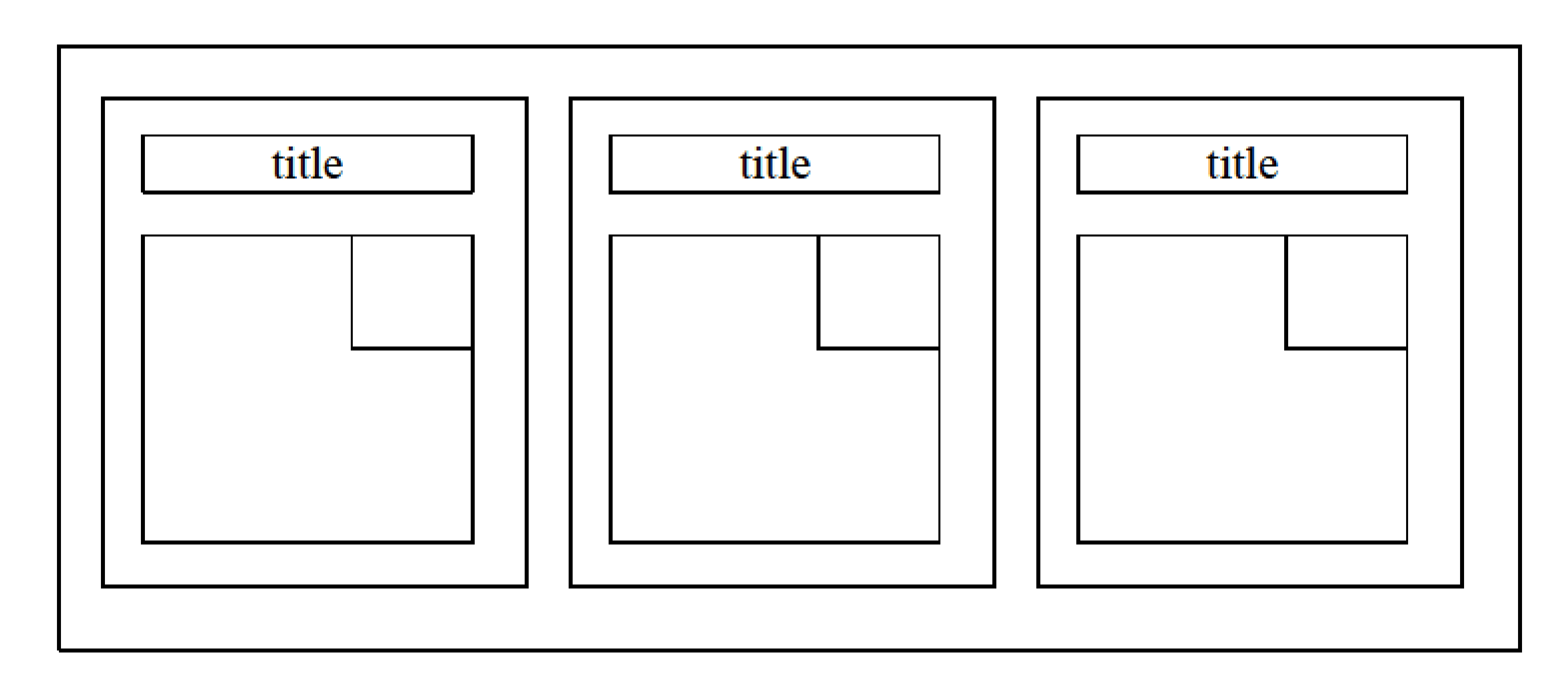


图 19-17　“新书上架”栏目的样式设计

因此，该模块的页面结构可以设计成如下代码所示：

```
<h2><img src="imgs/bullet2.gif" />新书上架</h2>
<div class="clear">
  <div class="show_prod">
    <a href="#">NetBeans</a>
    <div class="pic">
      <img src="imgs/new_icon.gif" class="icon" />
      <a href="#"><img src="pics/N07023858.jpg" class="prod_pic" /></a>
    </div>
  </div>
  <div class="show_prod">
    <a href="#">Ruby on Rail</a>
    <div class="pic">
      <img src="imgs/new_icon.gif" class="icon" />
      <a href="#"><img src="pics/N207022820.jpg" class="prod_pic" /></a>
    </div>
  </div>
  <div class="show_prod">
    <a href="#">SSH</a>
    <div class="pic">
      <img src="imgs/new_icon.gif" class="icon" />
      <a href="#"><img src="pics/N207024411.jpg" class="prod_pic" /></a>
    </div>
  </div>
</div>
```

这段代码的样式声明代码如下所示：

```
.show_prod {
    text-align:center;
    width:132px;
    margin:10px;
    float:left;
}
.show_prod a {
    color:#aaa;
    font-size:13px;
    text-decoration:none;
    line-height:21px;
}
.show prod .pic {
    background-image:url(../imgs/new prod box.gif);
    height:119px;
    position:relative;
}
.show_prod .pic .icon {
    position:absolute;
    top:0;
    right:0;
}
.show_prod .pic .prod_pic {
    margin:15px;
    width:60px;
    height:90px;
}
```

上面的样式表中声明了“新书上架”栏目中项目的宽度、文本对齐方式、边距、浮动等属性；然后设置项目中超链接的文本颜色、字号、文本修饰、行高等属性。再设置了项目中的图片区域的背景、高度、定位方式等属性(这里设置其相对定位其实对其位置并不产生任何影响，但是其子元素中需要以该对象为基准进行绝对定位，所以设置它的文本样式)。接着还设置了右上角新书图标定位方式及显示位置。最后设置图书图片的边距、宽度、高度等属性。

实现了上面样式的设置以后，在浏览器中运行的页面效果如图 19-18 所示。



图 19-18　“新书上架”栏目的运行效果

19.7 右侧辅助内容区域的详细设计

页面右侧辅助内容区域主要分为 5 个独立的栏目：“购物车”、“关于书店”、“优惠图书”、“图书分类”和“友情链接”。另外还有两个花式图案的栏目分隔条。下面分别对这些模块进行设计。

19.7.1 “购物车”栏目样式设计

“购物车”栏目非常简单，只有一个标题和一行购物车的简单信息。所以布局可以设计成如图 19-19 所示的结构。

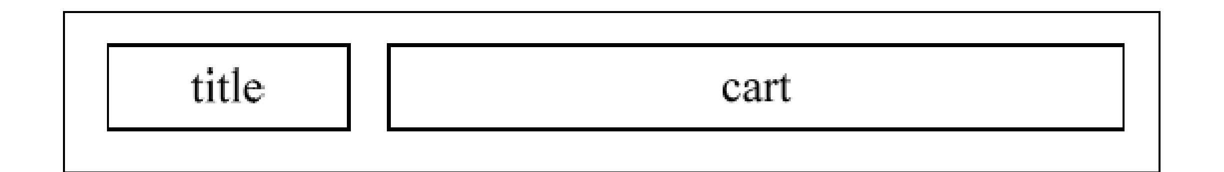


图 19-19 “购物车”栏目的设计布局

我们为该栏目设计的页面代码如下所示：

```
<div id="cart" class="clear">
  <h2><img src="imgs/cart.gif" width="21" height="18" />购物车</h2>
  <div class="info">2 件商品 | <span>总计：123￥</span><a href="#">查看购物车
</a></div>
</div>
<img src="imgs/border.gif" width="329" height="31" />
```

这里定义了一个购物车栏目的容器，其包含标题和购物车信息容器两个子元素。然后最下面的图片设置栏目下侧的分隔条。

上面 HTML 代码的 CSS 样式代码如下：

```
#cart h2 {
    float:left;
}
#cart .info {
    float:left;
    margin:20px 0 0 20px;
}
#cart .info span {
    color:#900;
}
#cart .info a {
    color:#900;
    margin-left:10px;
}
```

这里首先要设置栏目标题样式为左浮动；然后设置信息模块样式也向左浮动，并且设置了上边和下边的边距；再设置信息模块 span 标记的文本颜色为暗红色；最后设置信息模块中的超链接为暗红色，左边距为 10 像素。

设置完以后在浏览器中的运行效果如图 19-20 所示。

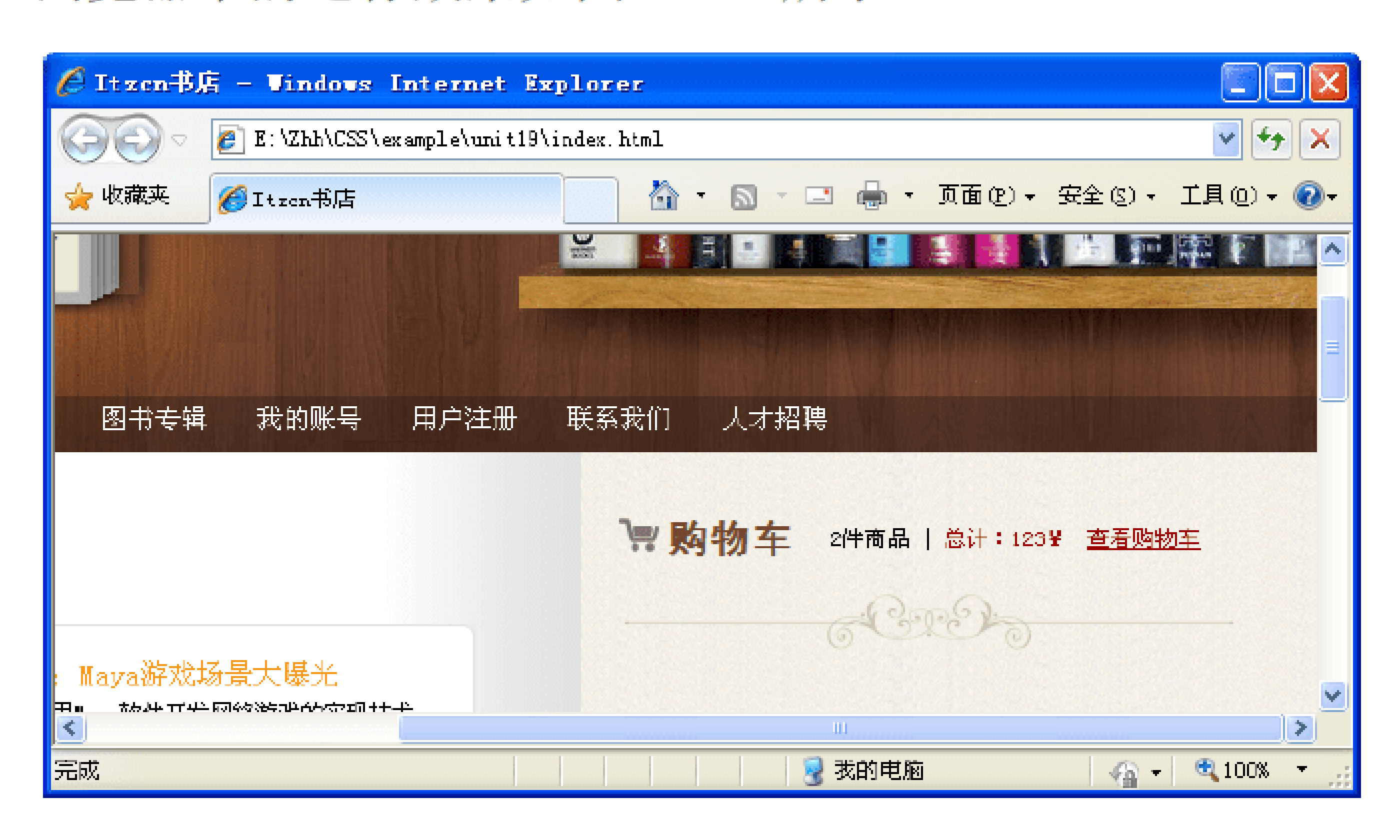


图 19-20　“购物车”栏目的设计效果

19.7.2　“关于书店”栏目样式设计

“关于书店”栏目的结构也非常简单，只需要一个容器，然后在容器中设置一个标题、一段文本和一张图片即可。所以页面结构可以设计成如图 19-21 所示的效果。

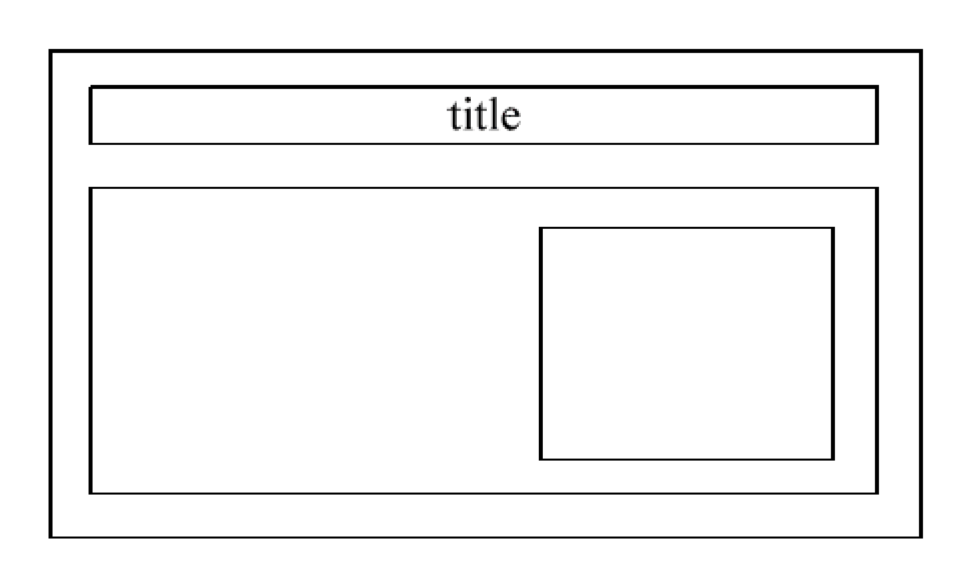


图 19-21　“关于书店”栏目的设计布局

我们为该栏目的页面编写的代码如下所示：

```
<div id="about_our">
  <h2><img src="imgs/bullet3.gif" width="26" height="25" />关于书店</h2>
  <div class="details clear">
    <img src="pics/about.jpg" />
    <p>    汇智科技公司关注 IT 教育培训市场的发展和变化，致力于在线教育，推进无纸化教学进程。
窗内网(www.itzcn.com)结合软件开发的实际需求，发布了 Flex、C#、Java、ASP.NET 和 JSP 等
系列视频教程。</p>
  </div>
</div>
<img src="imgs/border.gif" width="329" height="31" />
```

因为结构比较简单，也就不需要多解释了。最后一个 img 元素是栏目间的分隔线图片。

下面是为这段 HTML 代码添加的 CSS 样式代码：

```
#about_our .details {
    font-size:13px;
    line-height:18px;
    color:#333;
}
#about_our .details img {
    width:116px;
    height:80px;
    float:right;
    margin:10px;
}
```

代码看起来更加简单。标题样式不用设置，这里只设置文本字体、颜色、行高，图片的高度、宽度、右浮动、边距等属性。

在浏览器中页面的运行效果如图 19-22 所示。



图 19-22 “关于书店”栏目的运行效果

19.7.3 “优惠图书”栏目样式设计

“优惠图书”栏目只需要在栏目容器中添加一个标题和三个竖排的项目即可。每个项目中需要一个标题(书名)、一张图片和图片右上角的优惠标识。所以，该栏目的页面可以设计成如图 19-23 所示的结构。

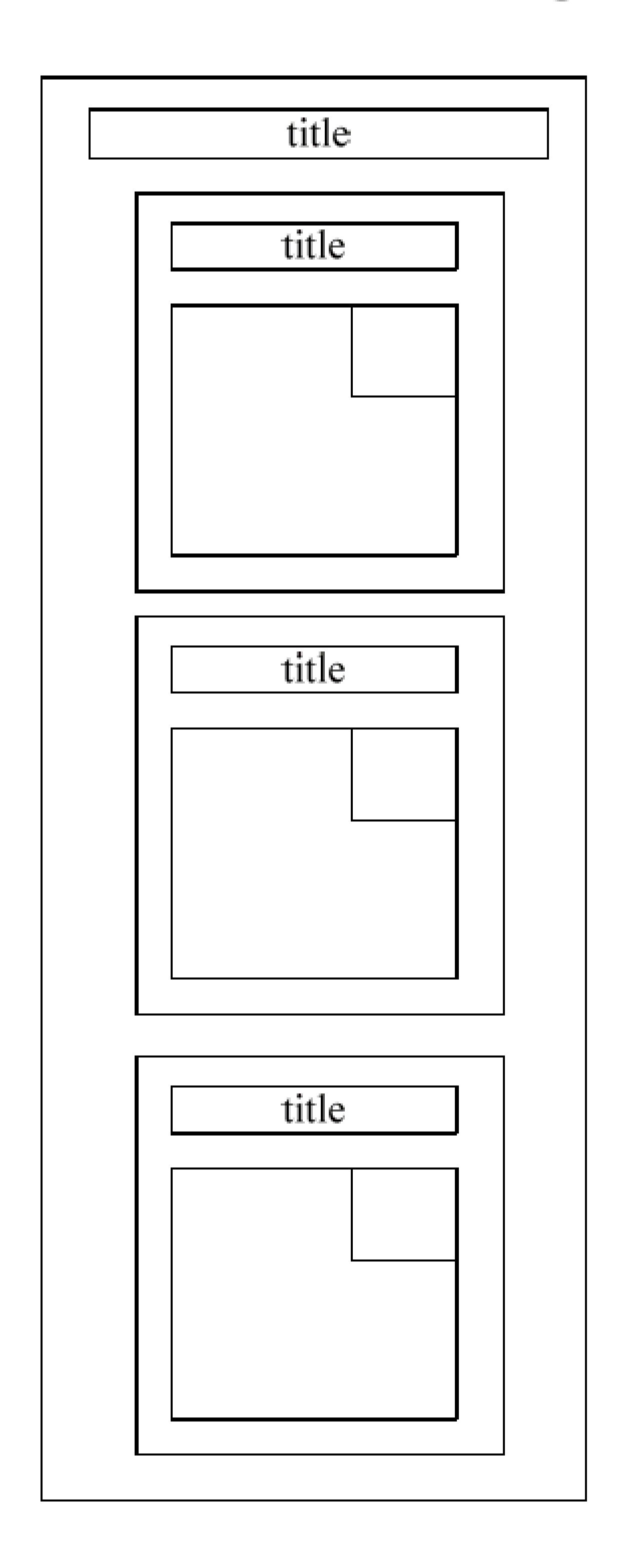


图 19-23 “优惠图书”栏目的样式设计

我们为该页面模块设计的结构代码如下所示：

```
<div id="promotions">
  <h2><img src="imgs/bullet4.gif" width="26" height="25" />优惠图书</h2>
  <div class="show_prod"> <a href="#">3ds Max 2009</a>
    <div class="pic">
      <img src="imgs/promo_icon.gif" class="icon" />
      <a href="#"><img src="pics/S528092300.jpg" class="prod_pic" /></a>
    </div>
  </div>
  <div class="show_prod"> <a href="#">3ds max 9</a>
    <div class="pic">
      <img src="imgs/promo_icon.gif" class="icon" />
      <a href="#"><img src="pics/S528093229.jpg" class="prod_pic" /></a>
    </div>
  </div>
  <div class="show_prod"> <a href="#">Premiere Pro 2.0</a>
    <div class="pic">
      <img src="imgs/promo_icon.gif" class="icon" />
      <a href="#"><img src="pics/S528093724.jpg" class="prod pic" /></a>
    </div>
  </div>
</div>
```

从上面的代码中不难发现，这里每一个项目都使用了在“新书上架”栏目中定义的项目样

式。因为它们显示的效果是完全相同的，这样就可以减少一些代码。这里我们只需要定义该栏目模块的样式即可。

下面是为该模块定义的 CSS 样式代码：

```
#promotions {
    width:152px;
    float:left;
}
```

这里只定义了该模块的宽度和左浮动属性。

在浏览器中运行以后，效果如图 19-24 所示。



图 19-24 “优惠图书”栏目的运行效果

19.7.4 “图书分类”栏目样式设计

从页面效果图中不难发现：“图书分类”栏目只有一个标题和一个列表。所以，该栏目可以设计成如图 19-25 所示的布局效果。

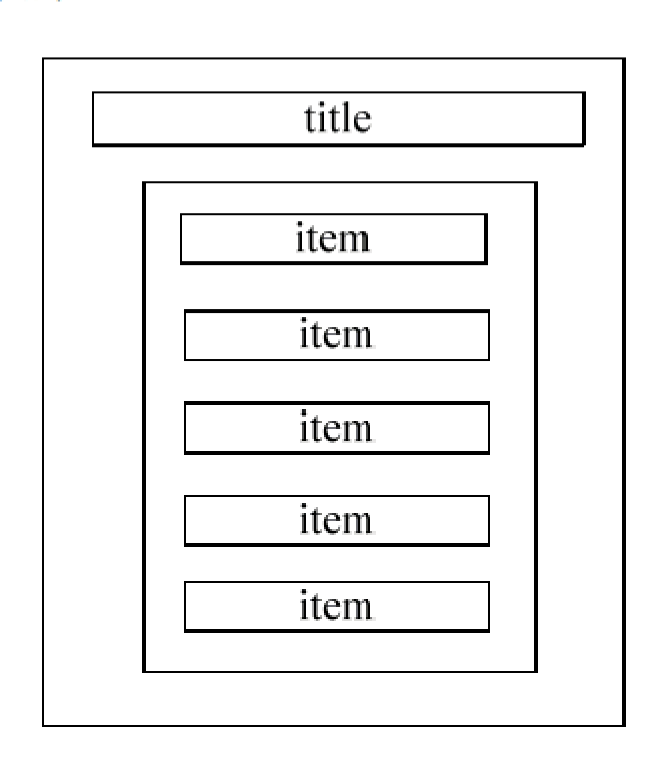


图 19-25 “图书分类”栏目的样式设计

我们为该页面模块设计的结构代码如下所示：

```
<div class="right_box">
  <h2><img src="imgs/bullet5.gif" width="26" height="25" />图书分类</h2>
  <ul class="list">
    <li><a href="#">三维动画</a></li>
    <li><a href="#">.NET 技术</a></li>
    <li><a href="#">Java 技术</a></li>
    <li><a href="#">PHP 技术</a></li>
    <li><a href="#">其它语言</a></li>
    <li><a href="#">Web 前端</a></li>
    <li><a href="#">数据库技术</a></li>
    <li><a href="#">基础应用</a></li>
  </ul>
</div>
```

上面这段代码非常简单，不作详细的解释了，下面是为这段代码编写的 CSS 样式代码：

```
.right_box {
    width:177px;
    float:right;
}
ul.list {
    padding:10px 0 0 50px;
}
ul.list li {
    list-style:url(../imgs/left_menu_bullet.gif);
    padding:3px 0;
}
ul.list li a {
    text-decoration:none;
    color:#000000;
}
ul.list li a:hover {
    text-decoration:underline;
}
```

在这段 CSS 样式代码中设置了该模块的宽度和右浮动属性、列表的填充属性、列表项的项目符号和填充属性、项目中超链接的文本颜色和文本修饰属性，以及列表中超链接的:hover 伪类的文本修饰属性。

设置完成以后，在浏览器中运行该页面，效果如图 19-26 所示。



图 19-26　“图书分类”栏目的运行效果

19.7.5 “友情链接”栏目样式设计

“友情链接”栏目的页面布局和 CSS 样式基本上与“图书分类”栏目完全相同，所以我们可以使用与“图书分类”栏目完全相同的页面布局和栏目样式。这里我们只需要编写页面代码即可。

为该栏目编写的页面代码如下所示：

```
<div class="right_box">
  <h2><img src="imgs/bullet6.gif" width="26" height="25" />友情链接</h2>
  <ul class="list">
    <li><a href="#">窗内网</a></li>
    <li><a href="#">窗内论坛</a></li>
    <li><a href="#">Google 搜索</a></li>
    <li><a href="#">百度搜索</a></li>
  </ul>
</div>
```

这里我们不用再为其添加样式。所以可以直接在浏览器中运行，效果如图 19-27 所示。

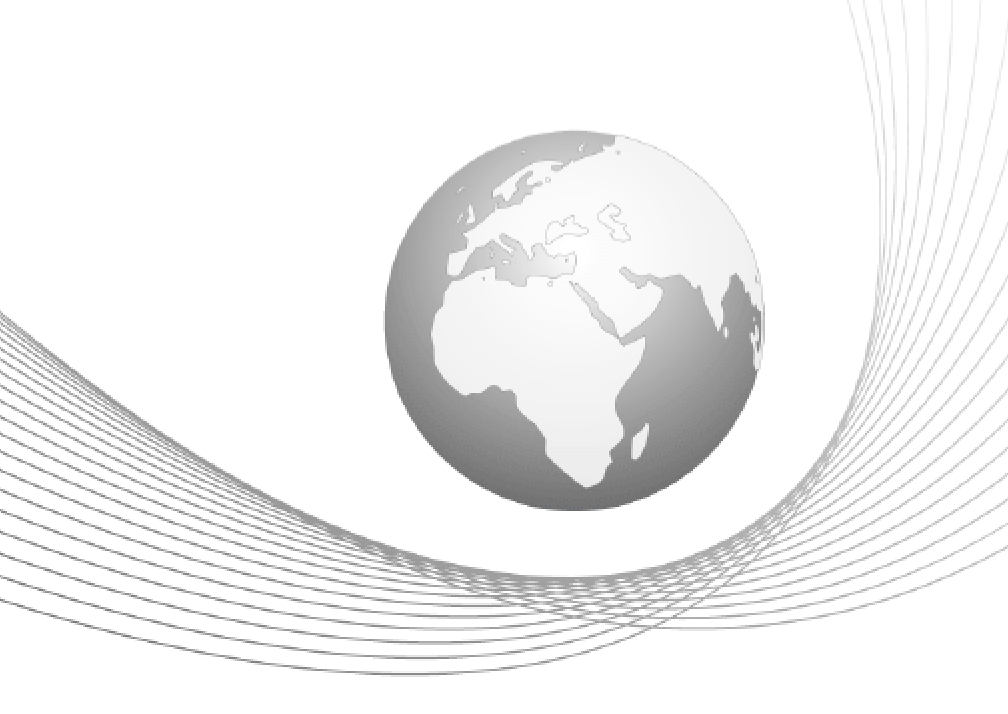


图 19-27 “友情链接”栏目的运行效果

19.8 总 结

在这个实例中，我们对前面学过的所有知识点进行了总结。比如设置文本的颜色、修饰、样式、字体，页面元素的背景、宽度、高度、边距、填充、边框，页面元素的定位、布局，页面元素的显示模式，等等。通过这个实例，我们对这些知识点进行了综合的运用，强化了理解，加深了记忆。

通过在这个项目中的实际操作，我们对整个页面布局和页面样式的设计有了更深层次的体会。不过这是一个非常简单的页面，所以设计起来没有太大的困难。对一门技术的掌握关键在于勤学多练，我们只有多做项目，不断地从实践中吸取经验，才能获得更大的收获。



第 20 章　设计微博首页

内容摘要：

微博是一个基于用户关系的信息分享、信息传播以及信息获取的平台，用户可以通过 Web、Wap 以及各种客户端组建个人社区，以 140 字左右的文字更新信息，并实现即时分享。

微博系统的出现，解决了 BBS 系统无法解决的“劣币驱逐良币”的问题，也解决了 Blog 系统中话题无法有效传播的问题，还解决了 SNS(Social Networking Services，社会性网络服务)系统虚假好友泛滥的社交问题，是目前最火爆的互动社区系统。

本章针对一个微博首页的结构进行分析，并通过 div + CSS 的 Web 标准方式对其进行实现。在讲解的过程中涉及了很多 CSS 设计技巧和兼容性解决方案。

学习目标：

- 了解什么是微博
- 了解微博的重要组成部分
- 熟悉 Dreamweaver 中站点的定义
- 掌握通过 CSS 定义 Logo 的方式
- 掌握 CSS 中列表样式的定义
- 掌握如何通过 CSS 将链接显示为图片
- 掌握浮动元素的清理方法
- 掌握微博各模块的实现方法

20.1 系 统 分 析

我们无论做什么事情，事先都需要做好充分的准备工作，要想设计出一个令自己和网友都满意的网站同样如此。在开始实际设计工作之前，我们首先要确定这样几个问题，然后才能开始实质性的工作：微博的含义是什么，都由哪些部分组成，需要呈现的又是什么，如何为这些内容合理地安排页面表现方式。

20.1.1 最终效果预览

微博是一种互动及传播性极快的交流工具，其传播速度甚至比传统媒体和网络媒体还要快。

微博所谓“微型博客(Micro Bloging)”是一种非正式的迷你型博客。它是一种基于 Web 2.0 技术的新兴的交流平台，是一种可以即时发布消息的类似博客的系统。它最大的特点就是集成化和开放化，你可以通过手机、IM 软件(gtalk、MSN、QQ、skype)和外部 API 接口等途径向自己的微博客发布消息。微博客的另一个特点还在于这个“微”字，一般发布的消息只能是只言片语，像 Twitter 这样的微博客平台，每次只能发送 140 个字符。

相对于强调版面布置的博客来说，微博的内容只是由简单的只言片语组成。从这个角度来说，对用户的技术要求门槛很低，而且在语言的编排组织上，没有博客那么高；只需要反映自己的心情，不需要长篇大论，更新起来也方便，和博客比起来，字数也有限制；微博开通的多种 API 使得大量的用户可以通过手机、网络等方式即时更新自己的个人信息。

在了解了有关微博的介绍之后，下面我们就来看看本实例中的最终运行效果，如图 20-1 所示。



图 20-1　最终微博首页效果

20.1.2　结构分析

在本节中，我们将会对图 20-1 所示效果进行拆分，然后分析它的具体组成结构。

首先抛开具体的网页内容，单从结构上看其实就是由我们所熟悉的上、中、下三部分结构组成。在本实例中从上至下依次命名为 login_top、login_body 和 footer 三部分，总体结构如图 20-2 所示。

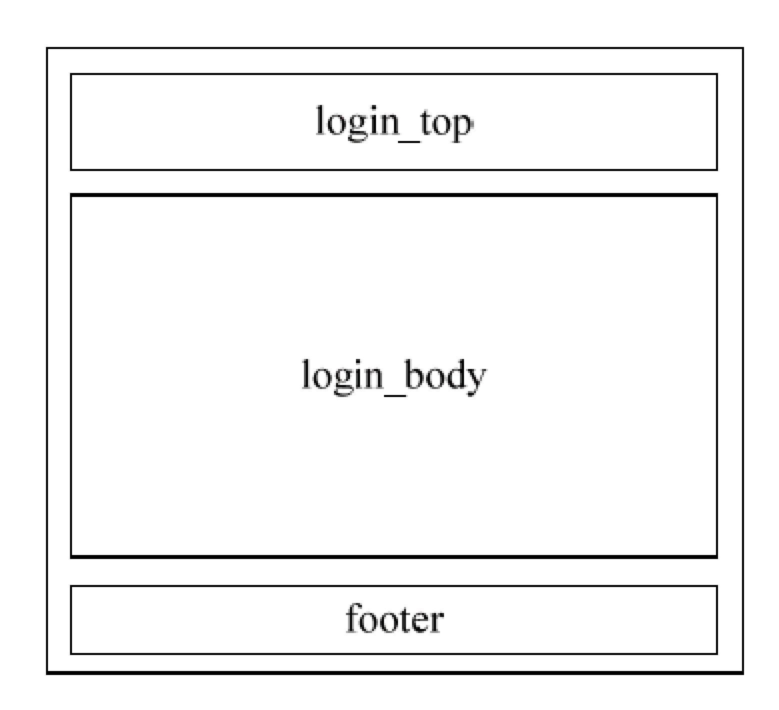


图 20-2　页面总体结构

login_top 块位于页面的最顶部，用于显示站点的 Logo，其结构如图 20-3 所示。

Login-body 块是页面的主要内容显示区域，在这里又分为左边展示区域和右边展示区域，如图 20-4 所示。

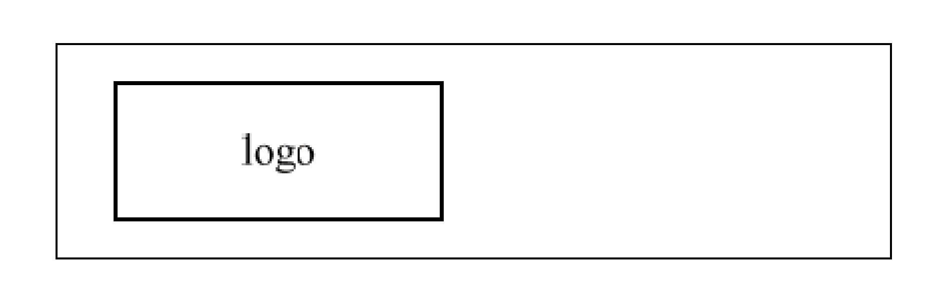


图 20-3　顶部结构

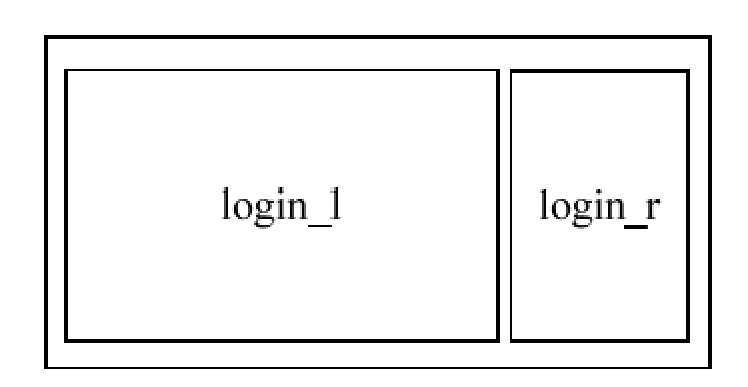


图 20-4　中部结构

左边的 login_l 块用于显示网站宣传标语、热议话题以及最新话题；右边的 login_r 块中则是微博注册按钮、登录表单以及最受关注的用户。细分这些内容后的结构如图 20-5 所示。

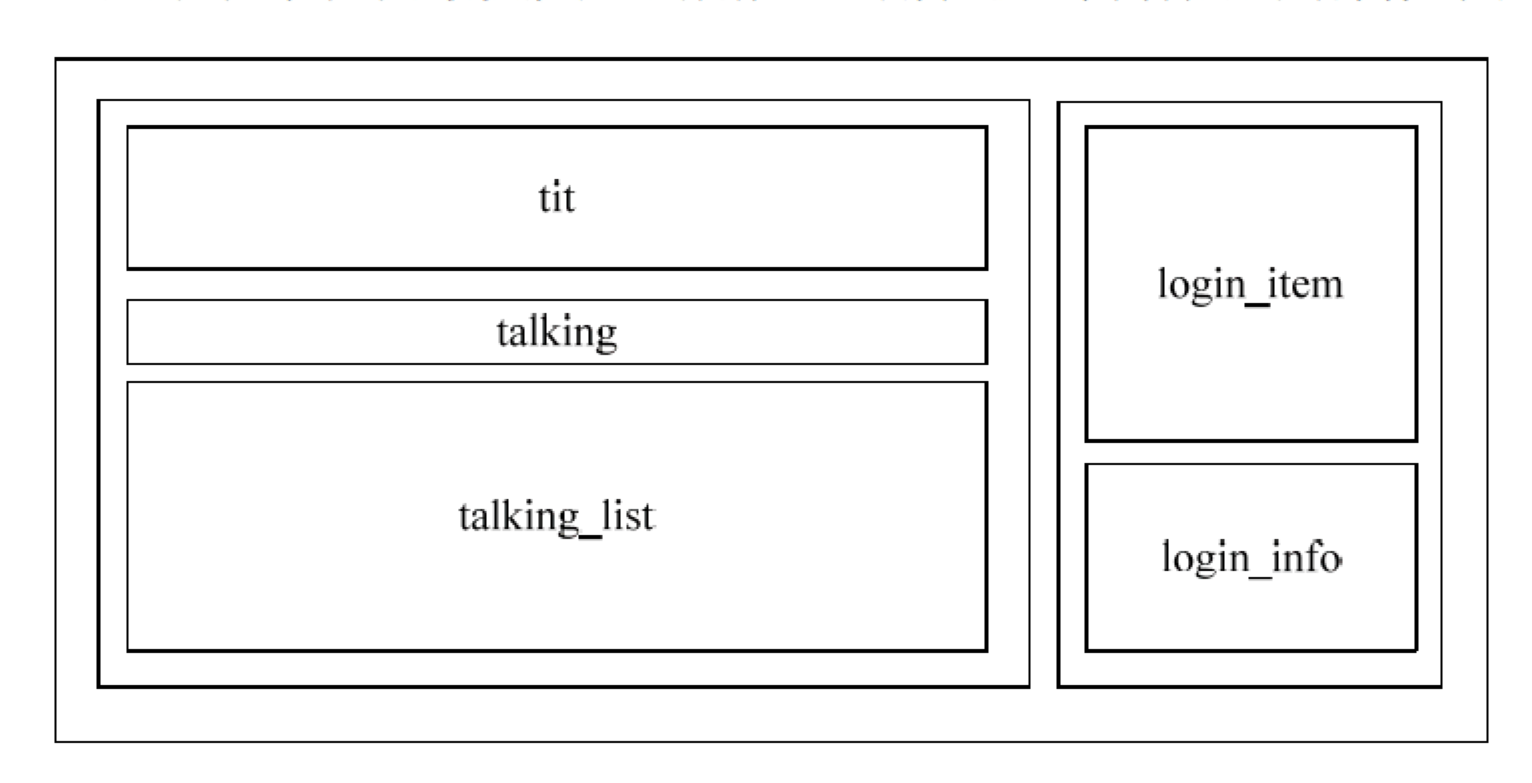


图 20-5　细分后的中部结构

关于页面底部的 footer 块，则比较简单，仅包含了一些文字连接，显示辅助信息和备案信息。所以这里就不再给出结构。

20.2 准 备 工 作

经过前面对系统结构的分析，明确了系统的模块划分。现在将开始着手准备网站的设计工作，这里我们使用的开发工具为 Dreamweaver CS4，开发浏览器为 IE 8。最终效果同时还在 Firefox 3、Chrome 和 Safari 等浏览器上进行了测试，未出现任何不兼容的问题。

20.2.1 创建站点结构

Dreamweaver CS4 最强大的功能就是站点管理，许多功能必须在创建站点后才能实现。因此，通常在 Dreamweaver CS4 中制作网站时，首先需要定义一个本地站点。

定义本地站点，就是在自己计算机的硬盘上建立一个目录，然后将所有与制作网页相关的文件都存放在里面，以便进行网页的制作和管理。因此，站点就可以理解成同属于一个 Web 主题的所有文件的存储地点。

创建本地站点首先应在本地计算机的磁盘中建立相应的物理目录(即指定文件夹)。例如，在本地计算机的“本地磁盘(D:)”盘中，创建 t 文件夹，然后即可用 t 作为本地站点的根文件夹。

(1) 在 Dreamweaver CS4 中执行“站点” | “新建站点”命令，弹出“未命名站点 1 的站点定义为”对话框，这个对话框包括“基本”选项卡和“高级”选项卡。“基本”选项卡相当于一个建立站点的向导，将引导设计者一步步地设置站点，“高级”选项卡则用来直接设置站点的各个属性。

(2) 在对话框的“基本”选项卡中输入新建站点的名称“MicroBlog”，注意对话框的标题也将随着新站点名称的变化而变化为“MicroBlog 的站点定义为”，如图 20-6 所示。



图 20-6　定义站点名称



建议初学者先使用“基本”选项卡，熟悉以后再直接用“高级”选项卡进行设置。

(3) 单击“下一步”按钮，弹出如图 20-7 所示的对话框。在这个对话框中设置站点的服务器技术，这里是一个单选按钮组，可设置站点是否使用 PHP 或者 JSP 等动态网页技术。如果选择“是，我想使用服务器技术”单选按钮，则将出现一个选择具体服务器技术的下拉列表框，在下拉列表框中选择适合的选项即可。

(4) 单击“下一步”按钮，弹出如图 20-8 所示的对话框，在对话框上方的单选按钮组中选择编辑站点文件的方法，其中包括两个选项，在这里使用默认值。在下方的文本框中输入新建站点所在的目录，或者单击“浏览”按钮选择。



图 20-7　选择使用的服务器技术



图 20-8　选择站点的存储位置

(5) 单击“下一步”按钮，此时可以指定如何与远程服务器进行共享，在列表中选择“无”选项，如图 20-9 所示。

(6) 单击“下一步”按钮，弹出如图 20-10 所示的对话框，现在所有的设置都已完成，这里给出了所有设置的列表。单击“完成”按钮，即可完成新建站点的操作。

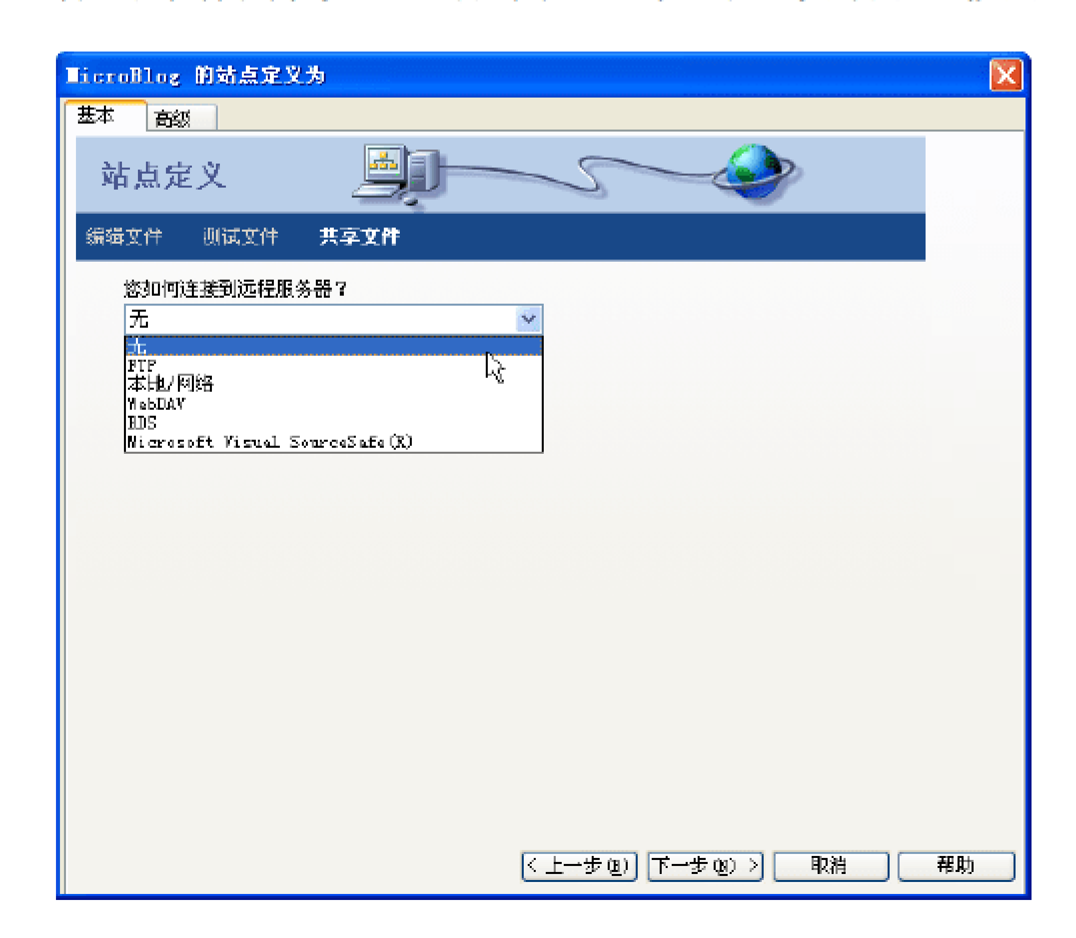


图 20-9　设置共享文件

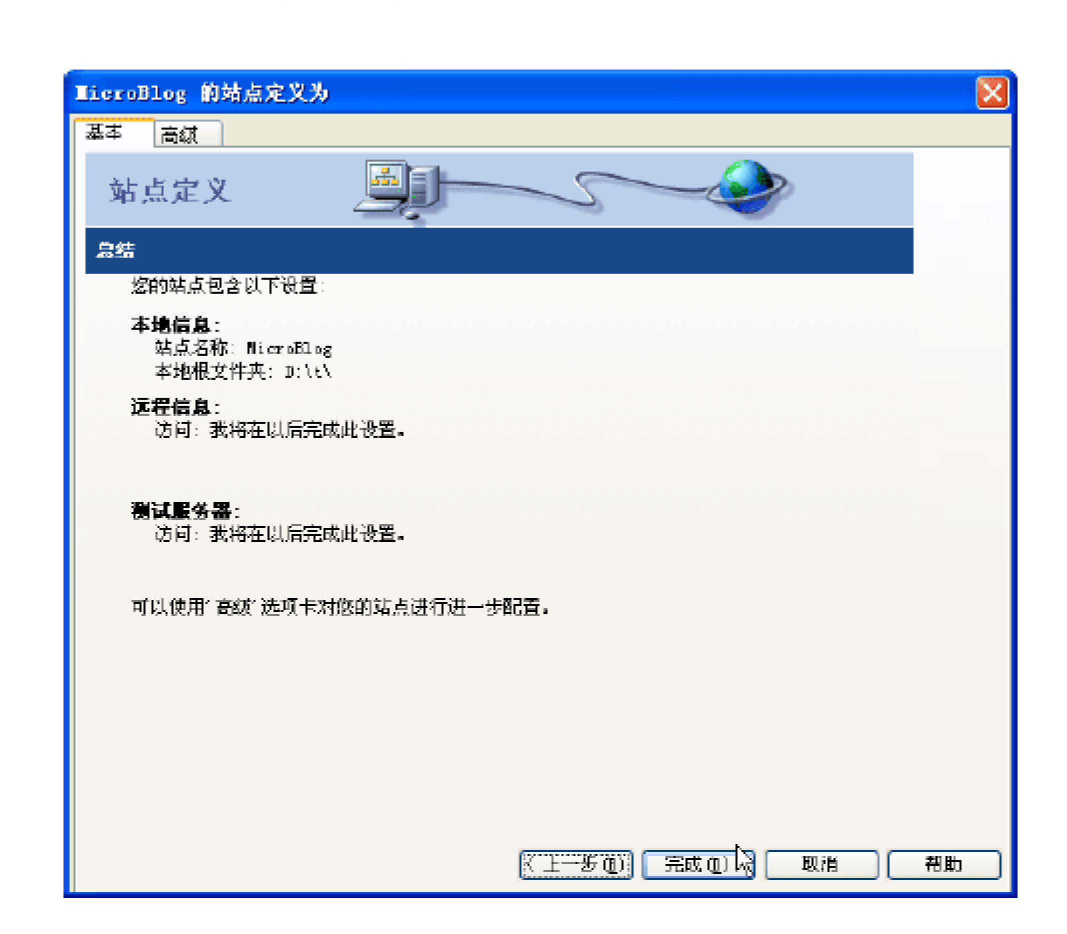


图 20-10　完成新建站点



如果本地的计算机中已安装 IIS 服务器，用户可以使用站点的定义将站点配置为动态站点，配置方法这里就不再进行介绍。

(7) 完成站点的创建，执行“窗口”｜“文件”是否该用引号呢？命令打开“文件”面板。在这个面板中显示了当前站点的目录结构，而且还可以执行各种管理操作。如图 20-11 所示为最终本实例的站点结构。

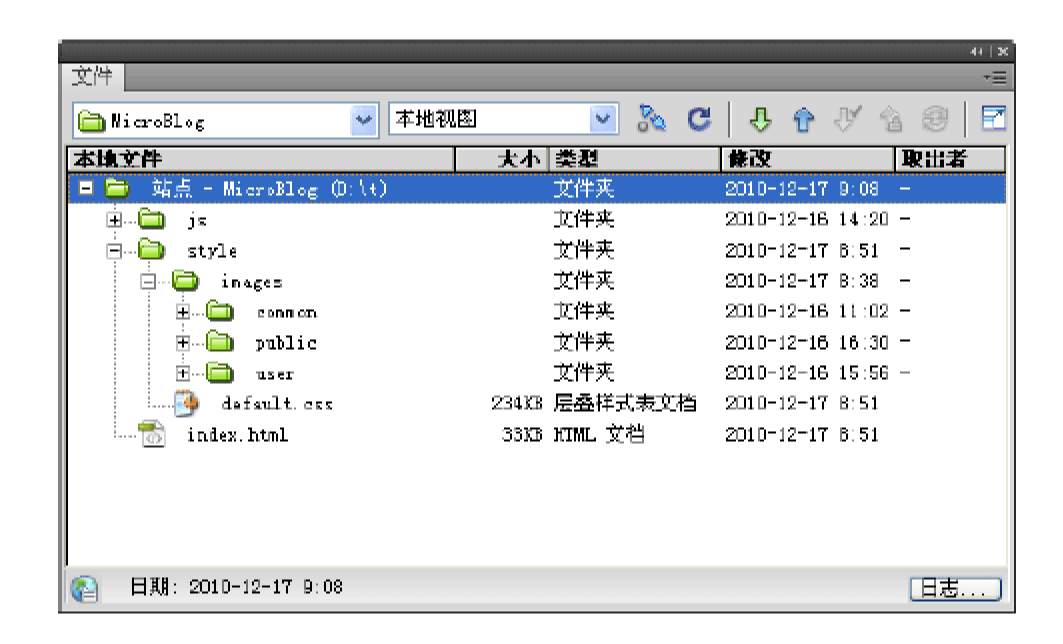


图 20-11　本地站点结构

实例中各个目录的含义如下。

- js 目录：保存站点中所用到的脚本文件，像 Java Script 代码等。
- style 目录：保存站点中所用到的 CSS 样式文件以及样式所带的图片资源。在该目录下的 images 子目录中又包含了 common、public 和 user 目录。

20.2.2　定义页面结构

打开 Dreamweaver CS4，执行“文件”|“新建”命令，弹出“新建文档”对话框。在左侧列表中选择“空白页”选项，“页面类型”列表中选择 HTML 选项，“布局”列表中选择“无”选项。从右侧“文档类型”下拉列表框中选择 XHTML 1.0 Transitional 选项，再单击“创建”按钮，如图 20-12 所示。

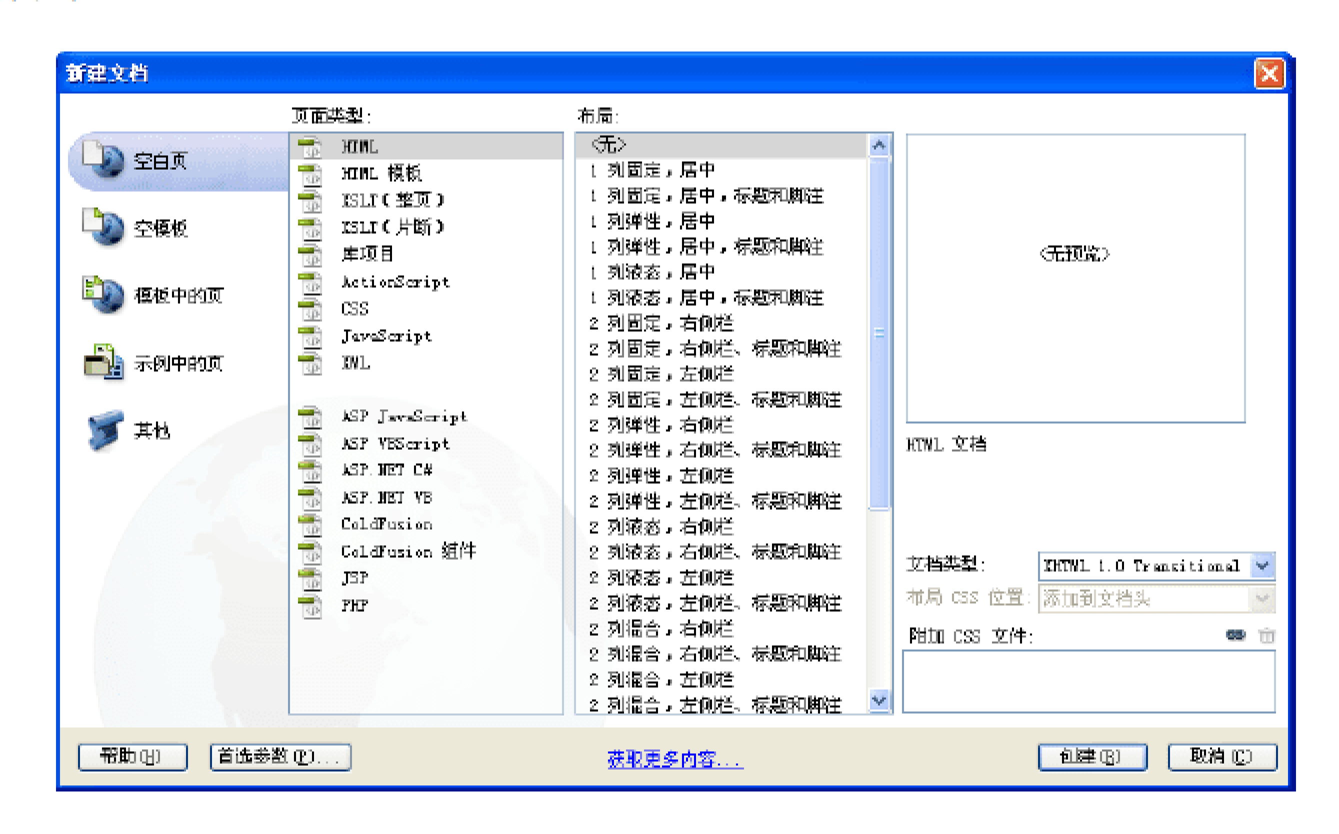


图 20-12　“新建文档”对话框

在“设计”视图中将标题修改为“记事狗微博-随时随地分享身边的新鲜事儿”。执行“文

件”|“保存”命令，在弹出的“另存为”对话框中，将新文件以“index.html”为名称保存到站点的根目录下。

接下来在站点的 style 目录下新建一个 CSS 文件，命名为“default.css”。返回 index.html 文件，在 head 元素中将该 CSS 文件链接入页面，代码如下所示：

```
<link href="style/default.css" rel="stylesheet" type="text/css" />
```

然后，在 index.html 中使用 div 元素并通过其 id 属性来定义整个页面的结构。如下所示为最终定义后的页面布局结构：

```
<div id="wrap">
  <div id="login_top">此处显示  id "login_top" 的内容</div>
  <div id="login body">此处显示  id "login body" 的内容
    <div id="login_l">此处显示  id "login_l" 的内容
      <div id="tit">此处显示  id "tit" 的内容</div>
      <div id="talking">此处显示  id "talking" 的内容</div>
      <div id="talking_list">此处显示  id "talking_list" 的内容</div>
    </div>
    <div id="login_r">此处显示  id "login_r" 的内容
      <div id="login_item">此处显示  id "login_item" 的内容</div>
      <div id="login_info">此处显示  id "login_info" 的内容</div>
    </div>
  </div>
  <div id="bodybt">此处显示  id "bodybt" 的内容</div>
</div>
```

由于各浏览器对所支持的 HTML 标签的默认样式存在不一致性，因此，要让显示效果相同，首先需要清除默认的样式，并使用新样式进行替换。如下给出的是本实例中需要替换的 CSS 样式，将其添加到 Style.css 中。

```
html {
    color:#000;
    font-family:Arial, Helvetica, sans-serif;
    font-size:12px;
}
body, div, label, dl, dt, dd, ul, ol, li, h1, h2, h3, h4, h5, h6, pre, code,
form, fieldset, legend, p, blockquote, th, td {
    margin:0;
    padding:0;
    border:0;
    outline:0;
}
body {
    color:#333;
    background:url(images/public/bdybg.jpg) top center no-repeat;
    background-color:#9edfe8;
}
```

保存对文件的上述修改。此时，在浏览器中预览运行效果，如图 20-13 所示。

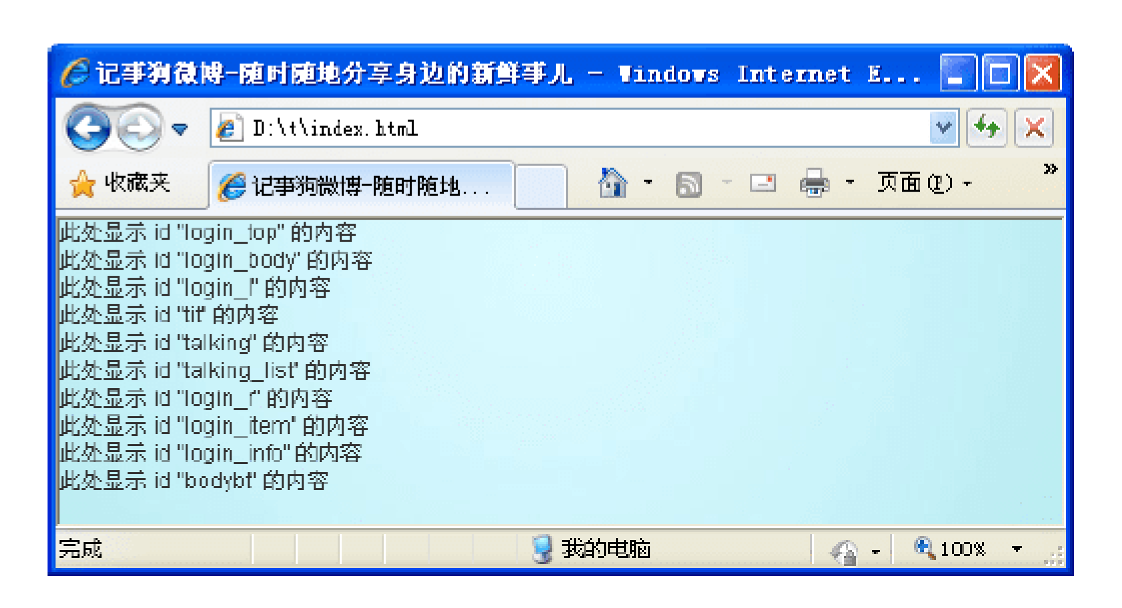


图 20-13　带页面结构的运行效果

20.3　设计顶部 Logo 展示模块

在页面的顶部区域中显示了微博网站的 Logo 图片，这个区域位于 body 的最上方，在显示时将非常醒目。

这部分在 HTML 页面结构中对应的是 login_top 模块，而该模块又被包含在一个名为 wrap 的主容器内。wrap 容器表示是整个页面，后面所有的内容也都包含在该容器内。

我们先来为主容器 wrap 定义一些样式，像宽度和高度等，代码如下所示：

```
#wrap {
    width:956px;
    margin:0 auto;
    min-height:500px;
}
```

然后，再为顶部 login_top 定义样式，主要是高度和定位属性，代码如下所示：

```
#login_top {
    height:97px;
    position:relative;
}
```

接下来，删除在 wrap 和 login_top 容器内的多余文本。定义一个微博超链接，以及一个用于显示 Logo 的 div 元素。

```
<div id="login_top">
   <a href="" class="llink"></a>
   <div class="LogoLink"></div>
</div>
```

在这里超链接 a 的 class 是 llink，创建一个“#login_top a.llink”CSS 样式规则来匹配它，并编写下面的代码。

```
#login_top a.llink {
    display:block;
    position:absolute;
    width:215px;
    height:81px;
```

```
    z-index:10;
    top:15px;
    left:34px;
}
```

最后是在 LogoLink 中显示一张背景图片，再定位它的位置，以及是否重复。这部分代码如下所示：

```
#login_top .LogoLink {
    background:url(images/public/Logo1.png) no-repeat center;
    width:215px;
    height:81px;
    position:absolute;
    top:15px;
    left:34px;
    _background:none;
_filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(src='images/publ
ic/Logo1.png', sizingMethod='crop');
}
```

完成后，在浏览器中打开来测试应用样式后的顶部运行效果，如图 20-14 所示。

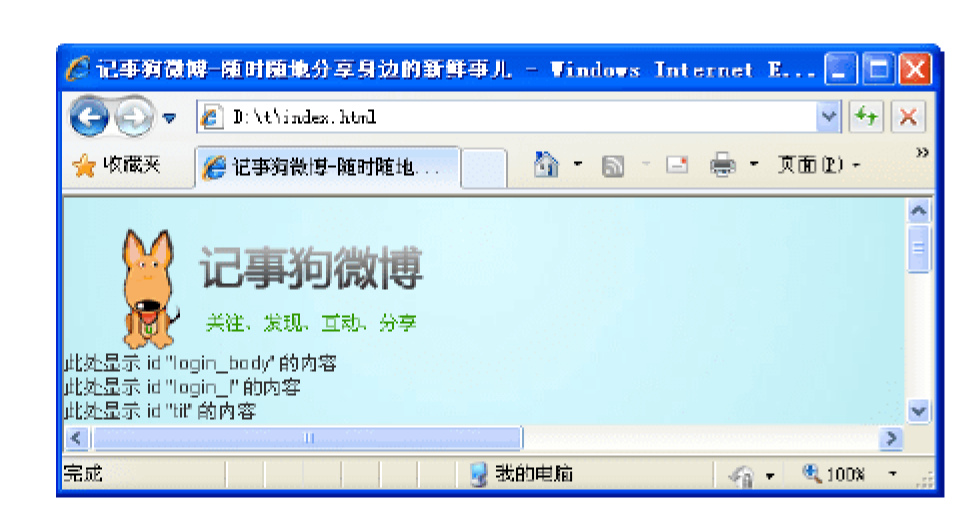


图 20-14　应用样式后的顶部运行效果

20.4　设计底部版权展示模块

现在，我们切换到 HTML 页面的最底部，在这里为显示版权信息的展示区域进行布局。

首先删除 bodybt 容器内的多余文本，再编写如下的 CSS 样式定义该模块的宽度、高度、位置和填充等。

```
#bodybt {
    background:url(images/public/bdybt.png) no-repeat;
    width:920px;
    height:54px;
    padding:20px 10px 0 26px;
    position:relative;
}
```

在 HTML 页面中为这一模块添加内容，包含一个带链接的导航菜单、客服电话、备案信息以及信息举报图片。最终代码如下所示：

```
<div id="bodybt">
    <div class="MIB_foot_new MIB_linkbr">
      <div class="lf">
        <p class="pl"><a  href="#">帮助</a><a  href="#">意见反馈</a><a
href="#">微博认证及合作</a><a href="#"  >开放平台</a><a  href="#">狗狗网导航
</a></p>
        <p>客服电话：400 560 5588  京 ICP 备 08104500 号 </p>
      </div>
      <div class="clearit"></div>
      <div class="botLogo"><a href="#"  ><img
src="style/images/public/botLogo.png" /></a></div>
    </div>
  </div>
```

如上述代码所示，使用分组选择符“MIB_foot_new MIB_linkbr”定义了一个 div 元素。在这个元素内，又可以分 lf、clearit 和 botLogo 三个部分。

如下所示是.MIB_foot_new 的 CSS 样式代码：

```
#bodybt .MIB_foot_new {
    color:#999;
    width:780px;
}
.MIB_foot_new {
    padding:0 20px 0 0;
}
#bodybt .MIB_foot_new a {
    color:#6eafd5;
}
```

页脚的这段布局代码中，使用 p 元素来进行分隔，其中 class 为 pl 的 p 元素中包含了网站辅助链接，再往下的 p 元素则显示了一段说明文字。

为了在第 1 个 p 元素中的链接之间产生间隔，这里使用 margin-right 设置了右边间隔。第 2 个 p 元素则设置了距底部的填充距离。

```
.MIB_foot_new .pl a {
    margin-right:11px;
}
.MIB_foot_new p {
    padding-bottom:5px;
}
```

再往下看看 botLogo 的 CSS 样式，由于它定义的图片需要固定在页面的右下角。所以，这里利用绝对定位，再通过设置距顶部和右侧的大小来实现，代码如下：

```
#bodybt .botLogo {
    position:absolute;
    right:15px;
    top:10px;
}
```

细心的读者可能会注意到，这里还有一个空 div 元素没有介绍。之所以放在最后介绍，是因为它的作用是避免产生两边浮动元素未闭合的情况，具体实现如下所示：

```
.clear, .clearit {
    clear:both;
    padding:0;
    margin:0;
}
```

类似这样的代码，我们会在后面布局时经常用到。如图 20-15 所示为应用样式后顶部和底部的运行效果。

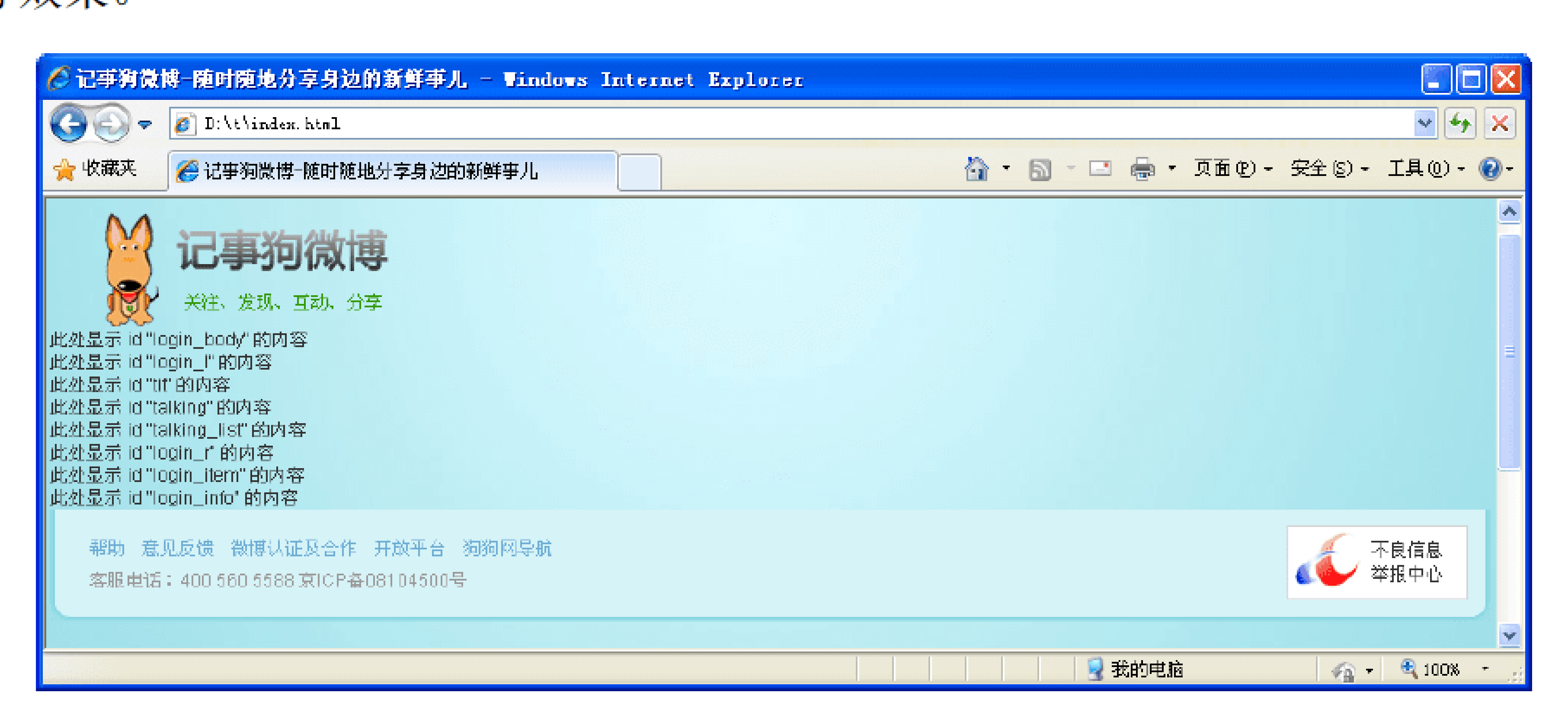


图 20-15　应用样式后顶部和底部的运行效果

20.5　设计左侧内容显示模块

左侧内容显示模块是由 login_body 容器定义的，它占用了页面中的大量展示区域，也包含了微博中最需要表现的内容。

开始具体布局与 CSS 设计之前，先删除在页面结构中定义的多余文本。

切换至 default.css 文件中，为 login_body 容器设置 CSS 样式，代码如下所示：

```
#login body {
    overflow:hidden;
    clear:both;
    position:relative;
    background:#fff url(images/public/mainbg.gif) repeat-y left top;
    height:820px;
}
```

在这里定义了一个高度，当内容超出这个高度值时则不显示，并且清除两边的浮动，定位方式为相对定位。同时还从 images/public/mainbg.gif 中定义了一个垂直重复的背景图片。

然后通过 login_l 容器来定义整个左侧的样式，包括宽度、浮动方式、是否填充和背景图片等。如下所示：

```
#login_body #login_l {
    width:683px;
    overflow:hidden;
    background:url(images/public/ml.jpg) no-repeat top;
    float:left;
    padding-left:3px;
}
```

20.5.1 宣传广告

微博的宣传广告紧接着站点 Logo 的下方，也是一个黄金位置。在这里通过一幅图片来吸引人的注意力。

这需要用 tit 容器的样式来完成，这部分代码如下所示：

```
#login_l #tit {
    background:url(images/public/tit.jpg) no-repeat;
    width:682px;
    height:164px;
    float:right;
    margin-top:13px;
    position:relative;
}
```

可以看到，定义的宽度和高度与背景图片的宽度和高度一致。采用相对定位，向右进行浮动，顶部有 13 像素的间距。如图 20-16 所示是应用样式后的运行效果。



图 20-16　运行效果

20.5.2 热议话题

热议话题这个栏目紧贴宣传广告栏目下方，是一个又长又窄的条幅。在这里显示了当前微博系统中用户议论最多的话题，每个话题都带有超链接，可以通过单击来查看该话题的详细内容。

如图 20-17 所示为这个模块的结构。通过#talking 可以定义这个区域的高度，并清除右侧的浮动，超出的内容隐藏不显示，代码如下所示：

```
#talking {
    height:36px;
    clear:right;
    overflow:hidden;
}
```

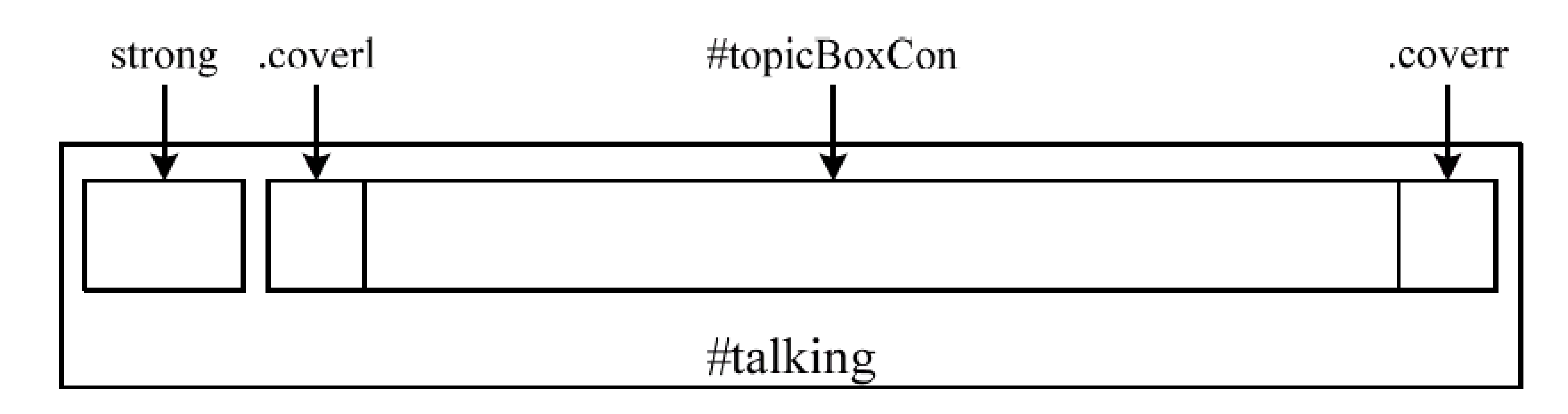


图 20-17　“热议话题”模块结构

在 talking 中添加一个 strong 元素并定义文字“正在热议”。

```
  <div id="talking">
    <strong class="">正在热议：</strong>
  </div>
```

接下来为 strong 元素定义样式，代码如下所示：

```
#talking strong {
    line-height:36px;
    height:36px;
    color:#cfc;
    padding:0 10px 0 45px;
    float:left;
}
```

运行后，strong 元素将向左浮动，显示在模块的最左侧，同时定义字体高度、行高、字体颜色和填充距离。

在 strong 元素之后插入两个 div，分别定义 class 为 coverl 和 coverr，代码如下所示：

```
<div class="coverl"></div>
<div class="coverr"></div>
```

从图 20-17 所示的结构中可以看到，coverl 和 coverr 是位于话题列表上的两个浮动块。它们实际就是利用两个半透明的 PNG 图片固定到那个位置，并通过其他 CSS 样式设置，达到淡入淡出的效果。

如下所示为所需的 CSS 样式：

```
#talking .coverl {
    pointer-events:none;
    _display:none;
    height:36px;
```

```
    width:50px;
    background:url(images/public/coverl.png) no-repeat 7px 0;
    z-index:10;
    left:110px;
    position:absolute;
}
#talking .coverr {
    pointer-events:none;
    _display:none;
    height:36px;
    width:50px;
    background:url(images/public/coverr.png) no-repeat -17px 0;
    z-index:10;
    left:636px;
    position:absolute;
}
```

前面的步骤为话题添加了一个标题以及两个遮罩块。现在，开始制作真正有关话题列表的布局。这部分代码如下所示：

```
<div class="acticle" id="topicBoxCon">
  <ul class="acticle_list" id="acticle_list">
    <li><a  href="#">下雪</a></li>
    <li><a  href="#">张宁海</a></li>
    <li><a  href="#">京东当当封杀门</a></li>
    <li><a  href="#">好冷</a></li>
    <li><a  href="#">NBA</a></li>
    <li><a  href="#">微访谈王煜全</a></li>
    <li><a  href="#">《让子弹飞》公映</a></li>
    <li><a  href="#">包身工</a></li>
  </ul>
</div>
```

这里使用一个 ul 列表，再定义一个 li 元素来表现一个话题。在 ul 列表中只需要关注话题内容，具体如何显示由外围的 div 来控制。

如下所示是控制位置的 CSS 样式：

```
#talking .acticle {
    overflow:hidden;
    width:549px;
    float:left;
    height:36px;
}
.acticle_list {
    overflow:hidden;
    height:36px;
    width:1200px;
    white-space:nowrap;
}
```

接下来的这一步则是定义 li 的位置，内部包含链接的颜色和鼠标移至上方时的颜色，代码如下所示：

```
.acticle_list li {
    float:left;
    margin-right:8px;
    line-height:36px;
}
#talking .acticle_list li a {
    color:#EAFFEA;
}
#talking .acticle_list li a:hover {
    color:#EAFFEA;
}
```

如图 20-18 所示为本节添加布局和 CSS 样式后，该栏目的运行效果。



图 20-18　“热议话题”栏目效果

20.5.3　会员列表和最新话题

会员列表和最新话题这两个模块是微博首页中占用面积最大的栏目，也是浏览者最感兴趣的栏目。

在会员列表模块中展现了由系统推荐的名人会员，以及随机产生的您可能感兴趣的会员，每位会员都显示了其头像与名称。最新话题模块则显示了系统所有会员当前发表的最新话题，每个话题都包含发表人头像、昵称、话题内容和发表时间。

根据开始的页面结构分析，我们知道这两个模块应该放在#talking_list 容器内。然后再划分出会员列表和最新话题模块，该模块的最终结构如图 20-19 所示。

在 index.html 中删除#talking_list 内多余的文本，然后根据图 20-19 所示来定义这个模块的 HTML 布局。代码如下所示：

```
<div id="talking_list">
   <div class="tList_mdu1">
      <!-- 此处为会员列表 -->
   </div>
  <div class="tList_mdu2">
      <!-- 此处为最新话题 -->
   </div>
```

```
    <div class="clearit"></div>
</div>
```

上述代码中最后一个 div 元素 clearit 用于清除上面两个模块产生的浮动元素问题。

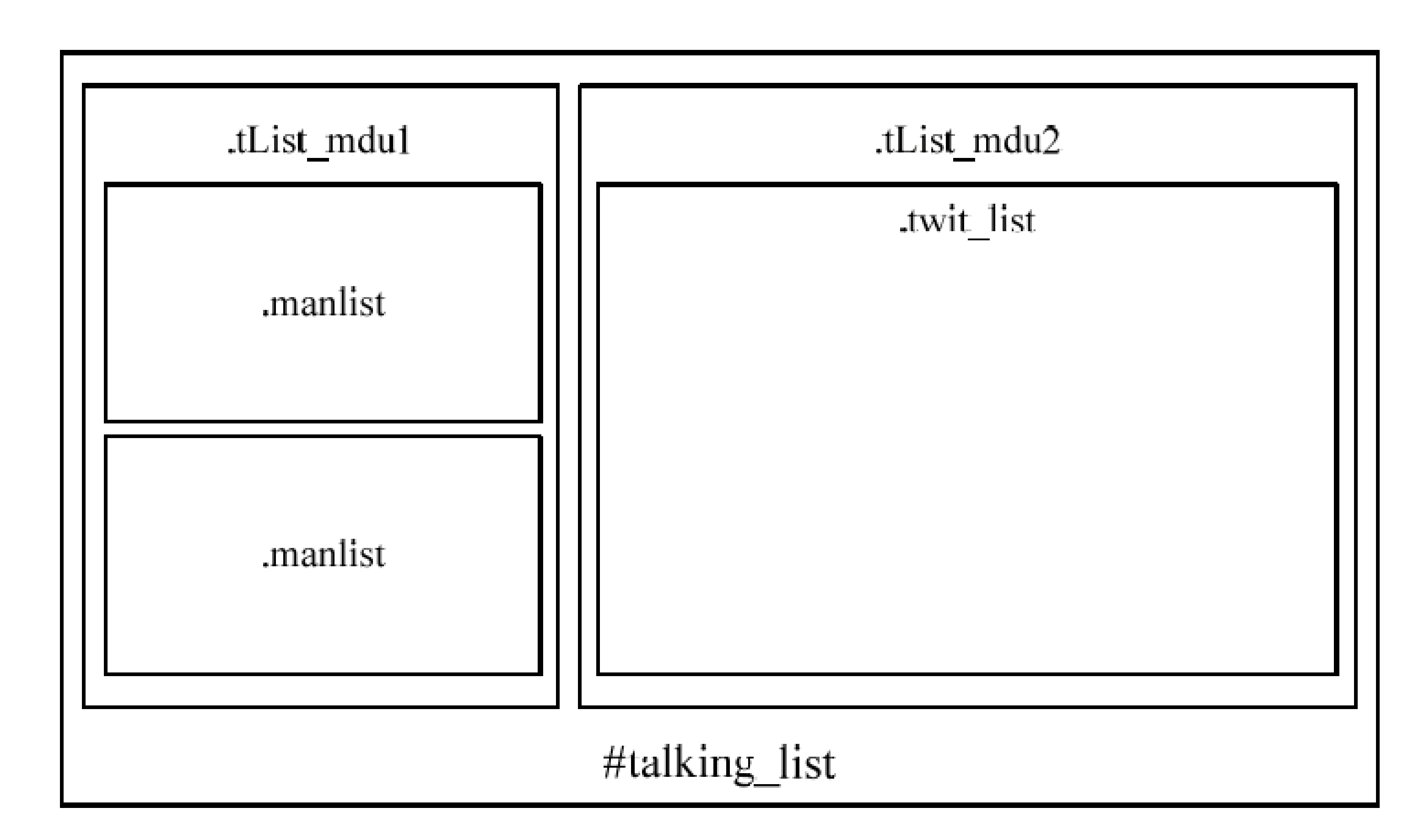


图 20-19　#talking_list 模块结构

接下来，开始为上述添加的 HTML 布局结构定义 CSS 样式。第一个是这里的主容器#talking_list，它的样式如下：

```
#talking_list {
    clear:both;
    padding-bottom:15px;
    padding-top:15px;
}
```

然后是内部包含的两个并列元素.tList_mdu1 和.tList_mdu2，它们所需的 CSS 样式如下所示：

```
.tList_mdu1 {
    width:224px;
    border-right:1px solid #EBEBEB;
    padding:0 0 0 24px;
    overflow:hidden;
    float:left;
}
.tList_mdu2 {
    height:570px;
    width:420px;
    overflow:hidden;
    position:relative;
    float:left;
}
```

至此，这两个栏目的框架就搭建好了。下面则需要向各自的容器中填充内容，再细化这些内容的表现方式。

按照从上至下，从左至右的顺序。首先制作会员列表中正在使用微博会员的列表，这部分的 HTML 布局如下所示：

```
<dl class="manlist">
  <dt>这些人正在用微博</dt>
  <dd>
   <ul class="userlist">
     <li><a href="#"  ><img src="style/images/user/1283136409.jpg"
/></a><a href="#" >祝红涛</a></li>
      <li><a href="#"  ><img src="style/images/user/1284457797.jpg"
/></a><a href="#"  >侯艳书</a></li>
      <li><a href="#"  ><img src="style/images/user/1290884345.jpg"
/></a><a href="#" >贺强</a></li>
      <li><a href="#"  ><img src="style/images/user/1283203478.jpg"
/></a><a href="#"  >陈锦伟</a></li>
      <li><a href="#"  ><img src="style/images/user/1291184181.jpg"  /></a><a
href="#"   >张振华</a></li>
      <li><a href="#"  ><img src="style/images/user/1291278034.jpg"
/></a><a href="#"   >王川川</a></li>
      <li><a href="#"  ><img src="style/images/user/1256134612.jpg"
/></a><a href="#"   >小喇叭</a></li>
      <li><a href="#"  ><img src="style/images/user/1291869868.jpg"
/></a><a href="#"   >朱凯</a></li>
      <li><a href="#"  ><img src="style/images/user/1283203880.jpg"
/></a><a href="#"   >平安北京</a></li>
      </ul>
    </dd>
</dl>
```

上述代码应该放在.tList_mdu1 容器内。不过需要注意的是，这里的层次结构与前面介绍过的有些不同。.manlist 容器表示是一个会员列表，在 dt 元素中定义了该列表的标题，例如这里的“这些人正在用微博”。

然后下面的.userlist 容器是真正的会员列表区域，其中每个 li 元素中有一个会员头像链接和会员昵称链接。

按照惯例从外至内依次定义 CSS 样式。整个会员列表.manlist 容器的样式如下：

```
.manlist {
    width:224px;
    overflow:hidden;
    float:left;
    display:inline;
    margin-bottom:15px;
}
```

如上述代码所示，这个会员列表的宽度为 224 像素，向左进行浮动实现紧贴左边排列，“display:inline”样式规则使列表内显示为块的元素，变为行内显示，即行内内联模式。

现在来完善该模块内其他布局的 CSS 样式，包括 dt 元素、li 元素、img 元素和超链接。最终代码如下所示：

```
.manlist dt {
    background:url(images/public/crown.png) no-repeat left 4px;
    font-weight:bold;
    line-height:22px;
    padding:0 0 8px 22px;
    border-bottom:1px solid #ebebeb;
    margin-right:24px;
}
.manlist .userlist li {
    float:left;
    text-align:center;
    padding:11px 0 0 0;
    width:60px;
    margin-right:10px;
    overflow:hidden;
    text-align:center;
    height:70px;
}
.manlist .userlist li a {
    color:#6eafd5;
    white-space:nowrap;
    display:block;
}
.manlist .userlist li img {
    display:block;
    margin:0 auto 4px;
    width:50px;
    height:50px;
}
```

至此，关于第一个会员列表的布局和 CSS 样式就都制作完成了。第二个会员列表紧贴第一个显示，同样使用的是.manlist 容器，其中的内容结构与 CSS 样式都相同，所以这里就不再重复介绍。

结束两个会员列表的制作，现在轮到最新话题模块了。该模块位于.tList_mdu2 容器下，具体的 HTML 布局结构如下所示：

```
<dl class="twit_list">
  <dt>大家正在说</dt>
  <dd>
    <div class="bottomcover"></div>
    <div class="maincontent" id="txtBoxCon">
     <!-- 在此处添加最新发表的话题列表 -->
    </div>
  </dd>
</dl>
```

可以看到代表整个话题列表的是.twit_list，在它的样式中需要定义列表宽度、浮动方向以及是否有间距等。本实例中使用的代码如下：

```
.twit_list {
    margin:0 0 0 27px;
    width:394px;
    overflow:hidden;
    float:left;
    display:inline;
}
```

我们还看到 dt 元素、dd 元素、.bottomcover 容器和.maincontent 容器，这 4 个块的 CSS 样式代码如下所示：

```
.twit_list dt {
    background:url(images/public/bubble.png) no-repeat left 4px;
    font-weight:bold;
    line-height:22px;
    padding:0 0 8px 22px;
    border-bottom:1px solid #ebebeb;
}
.twit_list dd {
    overflow:hidden;
}
.twit_list .maincontent {
    overflow:hidden;
}
.twit_list .bottomcover {
    position:absolute;
    bottom:0;
    background:url(images/public/halftransp.png) no-repeat 0 -5px;
    height:35px;
    width:394px;
    _background:none;
    _filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(src='images/
public/halftransp.png', sizingMethod='crop');
    pointer-events:none;
}
```

继续添加 HTML 布局，将光标定位到.maincontent 容器内，使用如下的代码来定义一个话题：

```
<div class="itemt">
  <div class="twit_item MIB_linedot2">
    <div class="twit_item_pic"><a href="#"  ><img
src="style/images/user/1289743382.jpg"    /></a></div>
    <div class="twit_item_content"> <a href="#"  title="酸酸珊">酸酸珊</a>: 某
人，己所不欲勿施于人，这个道理你应该懂的噻！下次请考虑一下别人的感受。
      <div class="twit item time">12 月 15 日 23:58</div>
    </div>
  </div>
</div>
```

即，每个 itemt 容器表示发表的一个话题，包含了发表人头像、昵称、话题内容和发表时间。重复添加这个 itemt 容器从而形成多个话题组成的列表。

在上面的 itemt 容器内共有 5 个 CSS 样式需要定义，这部分代码如下所示：

```
.twit_list dd .twit_item {
    overflow:hidden;
    _zoom:1;
    font-size:12px;
    padding:10px 0;
}
.twit_list dd .twit_item_pic {
    float:left;
    width:66px;
    padding-top:2px;
}
.twit_list dd .twit_item_content {
    float:left;
    width:316px;
    color:#666;
    line-height:18px;
}
.twit_list dd .twit_item_content a {
    color:#6EAFD5;
}
.twit_list dd .twit_item_time {
    float:left;
    width:316px;
    color:#999;
    line-height:20px;
}
```

经过前面漫长的步骤和 CSS 样式的逐步完善，最终完成了会员列表和最新话题两个模块的制作。现在在浏览器中运行，会看到如图 20-20 所示的效果。



图 20-20 运行效果

20.6 设计右侧边栏模块

从本节开始，我们将来完成微博首页中的最后一个模块，即位于页面右侧的边栏。

与上一节介绍的模块相比，这个边栏虽然比较窄，但是显示的内容也非常重要，像如何注册一个微博、登录到微博系统或者查看最流行的关注排行榜等。

表示整个右侧边栏的是#login_r 容器，它的 CSS 样式定义如下：

```
#login_body #login_r {
    width:267px;
    height:auto;
    overflow:hidden;
    background:url(images/public/mr.png) no-repeat top;
    float:left;
    padding-right:3px;
}
```

可以看到，仍然是向左进行浮动，通过定义宽度、背景图片和右填充间距来控制位置。

如下所示是它包含的两个子容器#login_item 和#login_info 的样式：

```
#login_r #login_item {
    height:318px;
    padding-top:30px;
}
#login_r #login_info {
    background:#FFF;
    padding-top:20px;
}
```

其中#login_item 子容器中包含的是会员注册与登录入口，#login_info 子容器中包含的是一个会员关注度排行榜。

20.6.1 会员注册与登录入口

微博之所以这么流行，很大程度上取决于它对会员之间的交流、互动具有比较强的优越性。

因此，如果我希望成为微博中的一员，就必须先进行注册，然后便可以通过登录微博与其他会员进行沟通。

既然会员注册这么重要，需要把它放在一个比较醒目的位置，以方便浏览者使用。如下所示为本实例中的注册 HTML 布局代码：

```
<div class="regtop"> <a href="#"  class="regbtn"></a>
   <p>注册就有机会获得手机充值卡！</p>
</div>
```

代码非常简单明了。regtop 说明这是注册表单，regbtn 表示一个注册按钮，在 p 元素中提

供了一个简单的注册描述，更加容易理解。

与此对应的 CSS 样式就没有这么简单了，它需要精确控制注册表单的宽度、对齐方式、内容四周的间距。还有注册按钮所采用的背景图片、鼠标移到上面时的状态以及描述文字的字体颜色等。

最终确定的 CSS 样式代码如下所示：

```
#login_item .regtop {
    width:200px;
    text-align:center;
    margin:0 0 32px 30px;
}
#login_item .regtop p {
    padding-top:4px;
    color:#F00;
}
#login_item .regbtn {
    display:block;
    background:url(images/public/regbtn.png) no-repeat 0 0;
    width:196px;
    height:52px;
}
#login_item .regbtn:hover {
    background-position:0 -52px;
}
```

下面我们来看看会员登录的 HTML 布局设计，这些代码如下所示：

```
  <div class="loginform" id="login_form">
    <div class="input_bg">
      <input type="text" id="loginname"/>
    </div>
    <div class="input_bg">
      <input type="password" id="password"/>
      <input type="text" id="password text" style="display:none;"/>
    </div>
    <div class="setting"> <span class="lf">
      <input type="checkbox" class="labelbox" id="remusrname" checked />
      <label>下次自动登录</label>
      </span><a class="rt" href="#" target="_blank">找回密码</a> </div>
    <div class="signin"><a href="#" class="sp"
id="login_submit_btn"></a></div>
  </div>
```

在登录时需要提供一个账户和密码、登录按钮，另外还提供了下次自动登录和找回密码功能。

在登录表单的 HTML 布局设计代码中，需要涉及的样式比较多，像账户输入文本框、密码输入文本框、自动登录复选框、找回密码链接、登录按钮以及整体背景的控制等。

所以，这里就不再逐个介绍，其中重要的 CSS 样式如下所示：

```
#login_item .loginform {
    width:194px;
    margin:0 0 0 30px;
}
#login_item .input_bg {
    background:url(images/public/inputbg.png) no-repeat;
    width:194px;
    height:31px;
    margin-bottom:10px;
}
#login_item .input_bg input {
    border:none;
    margin:2px 0 0 4px;
    height:20px;
    line-height:20px;
    width:184px;
    color:#ababab;
}
#login_item .setting {
    color:#666;
    height:24px;
}
#login_item .setting a {
    color:#6eafd5;
}
#login_item .setting input {
    vertical-align:middle;
    _vertical-align:text-bottom;
    margin-right:4px;
    margin-right:0\9;
}
#login_item .setting .rt {
    line-height:24px\9;
}
#login item .signin {
    text-align:center;
    margin-top:16px;
}
#login_item .signin a.sp {
    display:inline-block;
    width:97px;
    height:30px;
    background:url(images/public/signin.png) no-repeat 0 0;
}
#login_item .signin a.sp:hover {
    background-position:0 -30px;
}
```

如图 20-21 所示为本节添加布局和 CSS 样式后，会员注册与登录入口的运行效果。



图 20-21　运行效果

20.6.2　会员关注排行榜

所谓会员关注排行榜，是指在当前微博中的所有会员，按照关注度(浏览量)从高到低顺序排列，列出前 10 位的会员信息。

这些信息应该添加到#login_info 容器内。根据前面介绍的会员列表和话题列表的经验，这里仍然采用 dl 元素中嵌套 ul 列表的形式来完成，其中每个 li 元素表示一个会员信息，显示了当前排行名次、会员昵称和关注度。

最终确定在本实例中的代码如下所示：

```
<div class="ptless" id="login_info">
  <div class="leftline">
    <dl>
      <dt class="hasline"></dt>
      <dd class="boardlist">
        <ul class="tLoginTop10List">
          <li><em class="num1"><strong></strong></em><span><a href="#" >祝红涛
</a></span><cite>4,334,860</cite></li>
          <li><em class="num2"><strong></strong></em><span><a href="#">侯艳书
</a></span><cite>4,183,816</cite></li>
          <li><em class="num3"><strong></strong></em><span><a href="#">张均焘
</a></span><cite>3,973,360</cite></li>
          <li><em class="num4"><strong></strong></em><span><a href="#" >刘晓桐
</a></span><cite>3,734,721</cite></li>
          <li><em class="num5"><strong></strong></em><span><a href="#" >谢玲
</a></span><cite>3,305,328</cite></li>
          <li><em class="num6"><strong></strong></em><span><a href="#" >贺瑞芳
</a></span><cite>3,201,802</cite></li>
          <li><em class="num7"><strong></strong></em><span><a href="#" >李玲
</a></span><cite>3,099,529</cite></li>
```

```
        <li><em class="num8"><strong></strong></em><span><a href="#" >黄正华
</a></span><cite>3,097,717</cite></li>
        <li><em class="num9"><strong></strong></em><span><a href="#">赵从林
</a></span><cite>2,845,960</cite></li>
        <li><em class="num10"><strong></strong></em><span><a href="#" >陈阳
</a></span><cite>2,828,813</cite></li>
      </ul>
    </dd>
  </dl>
 </div>
</div>
```

在移动终端和上网设备日益多样化的今天，一个网站的访问方式不止一种。对于微博系统来说，也是如此。因此，这里我们在排行榜的下方，添加了一个使用手机上微博的地址，HTML 代码如下所示：

```
<div  class="dlblank">
 <dl>
  <dt>如何用手机上微博</dt>
  <dd>
   <p>WAP 版微博地址：<cite>m.gougou.com</cite></p>
   <p class="more"><a  href="#">更多方式<em>&gt;&gt;</em></a></p>
  </dd>
 </dl>
</div>
```

上述 HTML 代码需要添加到 leftline 容器内，“关注排行榜”模块 dl 元素之外。用到的 CSS 修饰样式如下所示：

```
.dlblank {
    padding:25px 0 0 25px;
    _padding:25px 0 0 32px;
    clear:both;
    height:120px;
}
.dlblank dt {
    background:url(images/public/phone.png) no-repeat left 1px;
    _line-height:19px;
    padding-left:22px;
}
.dlblank dd {
    padding-left:21px;
    line-height:21px;
}
```

至此，整个微博中重要模块的设计以及 CSS 的制作就完成了。如图 20-22 所示为本节添加布局和 CSS 样式后，会员关注排行榜的运行效果。



图 20-22　运行效果

20.7　总　　结

基于 Web 标准的网站设计的核心在于如何使用众多 Web 标准中的各项技术来实现表现与内容的分离，即网站的结构、表现、行为三者分离。只有真正实现了三者分离的网页设计，才是真正符合 Web 标准的网页设计。W3C(Web 标准的制订组织)建议网页设计者使用 XHTML 以更严谨的语言编写结构，并使用 CSS 来完成网页的布局表现，因此掌握基于 CSS 的网页布局方式，是实现 Web 标准的必备基础。

本章通过微博这个目前非常流行的主题展开讲解。首先，阐述了什么是微博，并给出一个微博的最终运行效果。然后对其进行结构分析，并逐步按照从上到下、从左至右的顺序依次来实现。

在本章介绍的过程中，遵循 Web 开发的习惯。先设计 HTML 代码，再定义 CSS 样式，而且在各模块的分析时尽可能用最少的 CSS 代码来解释。因此，并没有介绍所有 CSS 代码，有兴趣的读者可以查看提供实例的完整源代码。

附录　参 考 答 案

第 1 章

1. 填空题

(1) Cascading Style Sheet
(2) 蓝
(3) 27
(4) 绿

2. 选择题

(1) D
(2) A
(3) B
(4) D
(5) B

第 2 章

1. 填空题

(1) *
(2) table{color:red}
(3) id
(4) 类选择符
(5) 群组选择符

2. 选择题

(1) BCD
(2) A
(3) B
(4) D
(5) A

第 3 章

1. 填空题

(1) a:hover

(2) li:first-child {color:red;}

(3) li:first-letter { color:red; font-size:18px;}

(4) :lang

2. 选择题

(1) A

(2) B

(3) A

(4) A

(5) B

第 4 章

1. 填空题

(1) oblique

(2) letter-spacing

(3) Font

2. 选择题

(1) D

(2) C

(3) B

第 5 章

1. 填空题

(1) text-align

(2) rt

(3) vertical-ideographic

(4) text-decoration

(5) uppercase

(6) 内容将在边界内换行

2. 选择题

(1) D
(2) C
(3) B
(4) C

第 6 章

1. 填空题

(1) 样式
(2) Border
(3) 最大

2. 选择题

(1) C
(2) A
(3) C
(4) B

第 7 章

1. 填空题

(1) padding-bottom
(2) 上下左右边距
(3) 100px

2. 选择题

(1) A
(2) A
(3) A
(4) ABCD
(5) B

第 8 章

1. 填空题

(1) 16
(2) RGB
(3) background-color
(4) background-image
(5) no-repeat
(6) background-image
(7) 居中

2. 选择题

(1) C
(2) D
(3) A
(4) A

第 9 章

1. 填空题

(1) list-style-type
(2) 实心圆点项目符号
(3) outside
(4) list-style-image

2. 选择题

(1) C
(2) A
(3) B
(4) D
(5) B
(6) C

第 10 章

1. 填空题

(1) scrollbar-face-color
(2) scrollbar-track-color
(3) scrollbar-arrow-color

2. 选择题

(1) C
(2) C
(3) B

第 11 章

1. 填空题

(1) absolute
(2) bottom
(3) clip
(4) block

2. 选择题

(1) B
(2) A
(3) D
(4) A

第 12 章

1. 填空题

(1) collapse
(2) 显示
(3) 单元格水平间距与垂直间距均为 10px

2. 选择题

(1) B

(2) B
(3) D
(4) C

第 13 章

1. 填空题

(1)display
(2)fieldset
(3)accesskey
(4)background-color
(5)
```
<style type=text/css>
    input:focus{
        border-color:blue;
        background-color:blue;
        color:white;
        }
    </style>
```

2. 选择题

(1) D
(2) A
(3) C
(4) B

第 14 章

1. 填空题

(1) Alpha
(2) 边缘光晕
(3) Light
(4) RevealTrans

2. 选择题

(1) C
(2) A
(3) A
(4) C

第 15 章

1. 填空题

(1) 代码检查器
(2) 缩进代码
(3) JavaScript
(4) JavaScript

2. 选择题

(1) C
(2) B
(3) D

第 16 章

1. 填空题

(1) 序言
(2) 属性
(3) 元素
(4) <HTML:STYLE>
(5) “标记名称” + “#” + “ID 属性值”
(6) <?xml-stylesheet href="xsl 样式表的 URL" type="text/xsl"?>

2. 选择题

(1) A
(2) B
(3) A
(4) A
(5) D

第 17 章

1. 填空题

(1) 布尔值
(2) ||

(3) undefined

2. 选择题

(1) D
(2) B
(3) A
(4) B

第 18 章

1. 填空题

(1) lte
(2) 行内块、绝对定位块
(3) overflow、宽度
(4) 过渡版本

2. 选择题

(1) D
(2) C
(3) B
(4) A
(5) D
(6) A